Workbook O Level & IGCSE Computer Science

P2 Practical Problem Solving and Programming

Inqilab Ruknuddin Patel

Topical Past Paper Questions

Lecture Notes

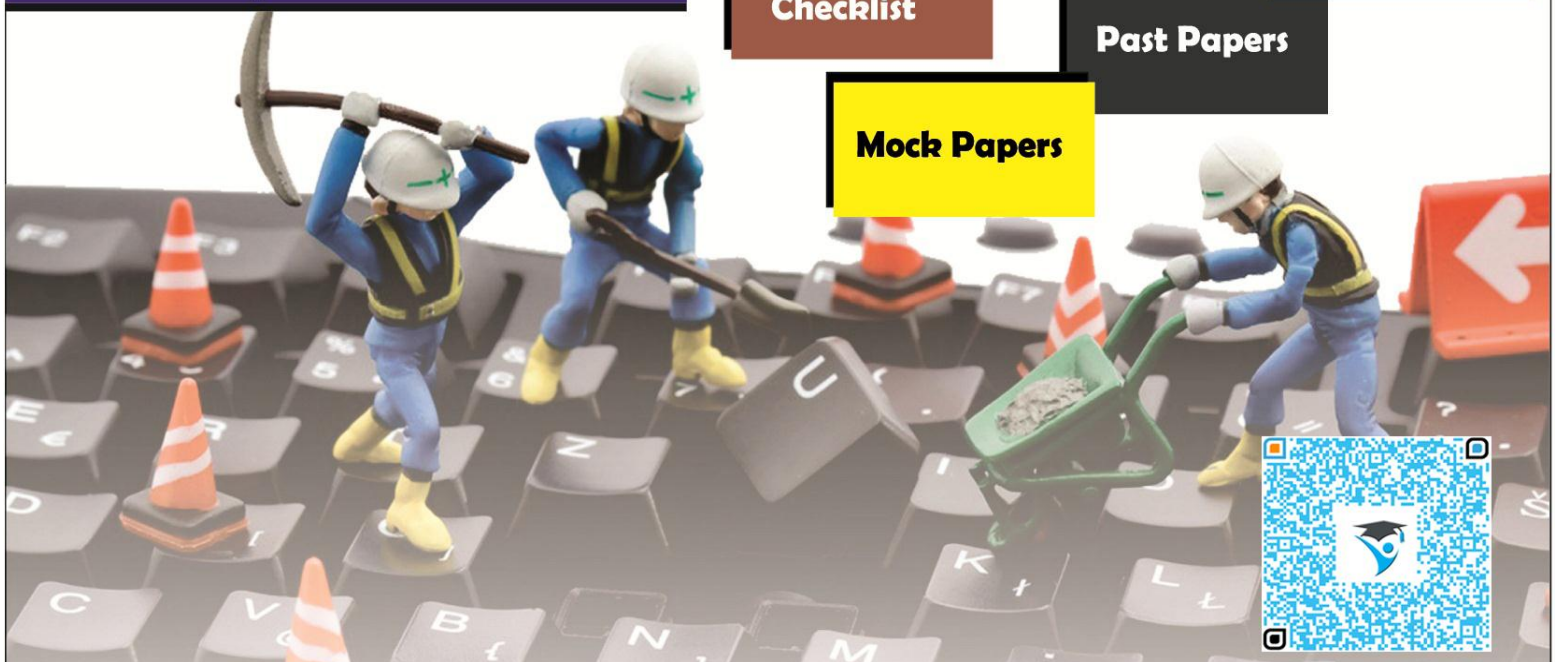
Practice Questions

Revision Guide

Revision Checklist

Past Papers

Mock Papers



Computer Science With Inqilab Patel

+923002724734 /inqilabpatel inqilab patel

inqilab inqilab-patel ruknuddin.com

Contents

Syllabus content & assessment at a glance

Sections	Topics
Section 1	**Theory of Computer Science** 1.1 Data representation 1.1.1 Binary systems 1.1.2 Hexadecimal 1.1.3 Data storage 1.2 Communication and Internet technologies 1.2.1 Data transmission 1.2.2 Security aspects 1.2.3 Internet principles of operation 1.3 Hardware and software 1.3.1 Logic gates 1.3.2Computer architecture and the fetch-execute cycle 1.3.3 Input devices 1.3.4 Output devices 1.3.5 Memory, storage devices and media 1.3.6 Operating systems 1.3.7High- and low-level languages and their translators 1.4 Security 1.5 Ethics
Section 2	**Practical Problem-solving and Programming** 2.1 Algorithm design and problem-solving 2.1.1 Problem-solving and design 2.1.2 Pseudocode and flowcharts 2.2 Programming 2.2.1 Programming concepts 2.2.2 Data structures; arrays 2.3 Databases

Assessment at a glance

Components	Weighting
Paper 1 Theory 1 hour 45 minutes This written paper contains short-answer and structured questions. All questions are compulsory. No calculators are permitted in this paper. 75 marks Externally assessed.	**60%**
Paper 2 Problem-solving and Programming 1 hour 45 minutes This written paper contains short-answer and structured questions. All questions are compulsory. 20 of the marks for this paper are from questions set on the pre-release material. 1 No calculators are permitted in this paper. 50 marks Externally assessed.	**40%**

About the developer of this workbook

Inqilab Patel is an O &A Level Computer Teacher. He highly qualified and experienced full of devotion and dedication. He has taught in many schools including Yaqeen Model School, Karachi Cadet School, KN Academy, Beacon House and The City School, PAF Chapter. **Cambridge** has selected him as a **Member** of **Cambridge Editorial Review Board** to review different course material, being uploaded from different parts of the world.

His entire career path revolves around computer science; either he was a student or a teacher. He got a chance to polish his skills of teaching and studying more about computers at various levels which has given him great confidence in presenting himself for any senior level position of transferring his knowledge to the youth.

He has not stopped, he is continuing with his education at the higher levels. It is his second semester of MPhil computer studies from a well-know university of Pakistan; The Institute of Business & Technology.

Inqilab Patel knows a lot of methods of teaching computers and has developed tutorial notes, worksheets and assignments for my students. He also maintains a website (www.ruknuddin.com) which is specifically designed for the support of those who want to excel in GCSE computer science. He also regularly contributes material to CIE teacher support website, for which he receives appreciation from different people across the world.

He has also received various training in innovative and special methods of teaching this subject.

Paper 2

Practical Problem-solving andProgramming

2.1.1 Problem-solving and design

- Show understanding that every computer system is made up of sub-systems, which in turn aremade up of further sub-systems
- Use top-down design, structure diagrams, flowcharts, pseudocode, library routines and subroutines
- Work out the purpose of a given algorithm
- Explain standard methods of solution
- Suggest and apply suitable test data
- Understand the need for validation and verification checks to be made on input data (validation could include range checks, length checks, type checks and check digits)
- Use trace tables to find the value of variables at each step in an algorithm
- Identify errors in given algorithms and suggest ways of removing these errors
- Produce an algorithm for a given problem (either in the form of pseudocode or flowchart)
- Comment on the effectiveness of a given solution

A system is a combination of parts or components, which work together to control a task or activity.

Computer System
A system which is made up of software, data, hardware, communications and people is considered as computer system.
Each computer system can be divided up into a set of sub-systems. Each subsystem can be further divided into sub-systems and so on until each sub-system justperforms a single action.

Computer system is often divided up into sub-systems. This division can be shown using top-down design to produce structure diagrams that demonstrate the modular construction of the system.Each sub-system can be developed by a programmer as sub-routine or an existing library routine may be already available for use. How each sub-routine works can be shown by using flowcharts or pseudocode.

- Top-down design
- Structure diagrams
- Flowcharts
- Pseudocode
- Library routines
- Sub-routines

1. Top-Down Design
Top-down design is the breaking down of a computer system into a set of subsystems, then breaking each sub-system down into a set of smaller sub-systems, until each sub-system just performs a single action.
This is an effective way of designing a computer system to provide a solution to a problem, since each part of the problem is broken down into smaller more manageable problems. The process of breaking down into smaller sub-systems is called ‘stepwise refinement’.
This structured approach works for the development of both large and small computer systems. When large computer systems are being developed this means that several programmers can work independently to develop and test different subsystems for the same system at the same time. This reduces the development and testing time.

2. Structure Diagrams

The **STRUCTURE DIAGRAM** shows the design of a computer system in a hierarchical way, with each level giving a more detailed breakdown of the system into sub-systems.

3. Flowcharts

A **FLOWCHART** shows diagrammatically the steps required for a task (sub-system) and the order that they are to be performed. These steps together with the order arecalled an **ALGORITHM**. Flowcharts are an effective way to communicate thealgorithm that shows how a system or sub-system works.

4. Pseudocode

PSEUDOCODE is a simple method of showing an algorithm, using English-like words and mathematical operators that are set out to look like a program.

5. Library routines

A **LIBRARY ROUTINE** is a set of programming instructions for a given task that isalready available for use. It is pre-tested and usually performs a task that isfrequently required. For example, the task 'get time' in the checking-for-the-alarmtime algorithm would probably be readily available as a library routine.

6. Sub-routines

A **SUB-ROUTINE** is a set of programming instructions for a given task that forms a subsystem, not the whole system. Sub-routines written in high-level programminglanguages are called 'procedures' or 'functions' depending on how they are used.

Test Data

Test data is the data that is used in tests of a software system.

In order to test a software application we need to enter some data for testing most of the features. Any such specifically identified data which is used in tests is known as test data.

There are following three types of test data:

1. Normal Data
2. Abnormal Data
3. Extreme Data

1. Normal Data

This is the data a computer system should work on.Testing needs to be done to prove that the solution works correctly. In order to do this a set of test data should be used together with the result(s) that are expected from that data. The type of test data used to do this is called **NORMAL DATA**, this should be used to work through the solution to find the actual result(s) and see if these are the same as the expected result(s).

For example, here is a set of normal test data for an algorithm to record the percentage marks from 10 end-of-term examinations for a student and find their average mark:

Normal test data: 50, 50, 50, 50, 50, 50, 50, 50, 50, 50

Expected result: 50

2. Abnormal Data

This is data that should cause the system to tell the user that there is a problem with data entered into the system.Testing also needs to be done to prove that the solution does not give incorrect results. In order to do this, test data should be used that will be rejected as the values are not suitable. This type of test data is called **ERRONEOUS** or **ABNORMAL TESTDATA**; it should be rejected by the solution.

For example erroneous/abnormal data for an algorithm to record the percentage marks from 10 end-of-term examinations for a student and find their average mark could be:

Erroneous/abnormal data:–12, eleven

Expected results: these values should be rejected

3. Extreme Data

When testing algorithms with numerical values, sometimes only a given range of values should be allowed. For example, percentage marks should only be in the range 0 to 100. The algorithm should be tested with **EXTREME DATA**, which, in this case, are the largest and smallest marks that should be accepted. Extreme data are the largest and smallest values that normal data can take.

Extreme data: 0, 100

Expected results: these values should be accepted

Rogue Values

A sequence of inputs may continue until a specific value is input. This value is called a **rogue value** and must be a value which would not normally arise.

A rogue value lets the computer know that a sequence of input values has come to an end.

Example

A number of marks are to be input (terminated by a rogue value of -1). How many of them are over 50?

```
Counter ←0
INPUT Marks
REPEAT
    IF Marks >50 THEN Above50 ←Above50 +1
    INPUT Marks
UNTIL Marks=-1
OUTPUT Above50
```

Validation and Verification

Validation and verification are two ways to check that the data entered into a computer is correct. Data entered incorrectly is of little use.

Data verification

Verification is performed to ensure that the data entered exactly matches the original source.Verification means checking the input data with theoriginal data to make sure that there have been notranscription errors (transcription means copyingthe data). The standard way to do this is to input thedata twice to the computer system. The computerthen checks the two data values (which should be thesame) and, if they are different, the computer knowsthat one of the inputs is wrong. E.g. entering password twice during sig-up.Verification methods include:

- double entry
- screen/visual check (proof reading)
- parity check
- Checksum.

Validation is an automatic computer check to ensure that the data entered is sensible and reasonable. It does not check the accuracy of data.

For example, a secondary school student is likely to be aged between 11 and 16. The computer can be programmed only to accept numbers between 11 and 16. This is a **range** check.

However, this does not guarantee that the number typed in is correct. For example, a student's age might be 14, but if 11 is entered it will be valid but incorrect.

Types of validation

There are a number of validation types that can be used to check the data that is being entered.

Validation type	How it works	Example usage
Range check	Checks that a value falls within the specified range	Number of hours worked must be less than 50 and more than 0
Length check	Checks the data isn't too short or too long. Values must be a specific length.	A password which needs to be six letters long
Limit Check	Similar to Range Check but the rule involves only one limit.	>=0 means reject negative numbers. Date of birth must not be later than a date.
Type Check	Checks that the data entered is of a given data type,	Number of brothers or sisters would be an integer (whole number).
Character Check	Checks that when a string of characters is entered it does notcontain any invalid characters or symbols,	A name would not contain characters such as %, and a telephone number would only contain digits or (,), and+.

Validation type	How it works	Example usage
Format Check	Checks the data is in the right format. Valuesmust conform to a specific pattern, for example, two letters followed by six digits followed by a single letter	A National Insurance number is in the form LL 99 99 99 L where L is any letter and 9 is any number
Presence check	Checks that data has been entered into a field	In most databases a key field cannot be left blank
Check digit	The last one in a code are used to check the other digits are correct	Bar code readers in supermarkets use check digits

Q1) Activity of data validation and verification:

1) What is an automatic computer check to make sure data entered is sensible and reasonable known as?

a) Double entry b) Verification c) Validation

2) What validation type would make sure a post code was entered in the correct format?

a) Length check b) Format Check c) Presence check

3) What validation type would you use to check that numbers fell within a certain range?

a) Range check b) Presence Check c) Check digit

4) What validation type checks that a field is not left blank?

a) Format check b) Length check c) Presence check

5) What validation type uses the last one or two digits to check the other digits are correct?

a) Length check b) Format check c) Check digit

6) What validation type checks a minimum number of characters have been entered?

a) Length check b) Format check c) Range check

7) Data is to be entered into a computer in the format YYMMDD. Which of the following is not a valid date?

a) 310921 b) 211113 c) 21st June 2004

8) Which of the following statements is false?

a) Validation can check that the data is sensible

b) Validation can check that the data falls between certain allowable boundaries

c) Validation can check that the data is correct

9) Which of the following is NOT a method of verification?

a) Double entry - typing the data in twice and getting the computer to check the second version against the first

b) Using presence, range and length checks to make sure that no mistakes happen
c) Printing out what you have typed in and comparing it against the source data

Q2) Summer 2014 pq11

A hospital holds records of its patients in a database. Four of the fields are:
• date of visit (dd/mm/yyyy)
• patient's height (m)
• 8-digit patient ID
• contact telephone number

The presence check is one possible type of validation check on the data. For each field, give another validation check that can be performed. Give an example of data which would fail your named validation check. A different validation check needs to be given for each field.

field name	name of validation check	example of data which would fail the validation check
date of visit		
patient's height		
patient ID		
Contact telephone number		

Q3) Summer 2013 P12

A company requests new customers who register online to give the following details:
• name
• address
• type of credit/debit card
• payment card number

All details must be entered.

(a) (i) Describe one suitable different validation check for each field.

Name: ……..
Address: ……..
type of credit/debit card: ……
payment card number: …… [4]

Q4) Summer 2012 P12

State two different validation checks and give an example of their use. Each example should be different.

Check 1: ……
Use: ……
Check 2: ……
Use: …… [4]

2.1.2 Pseudocode

2.1.2 Pseudocode
• understand and use pseudocode for assignment, using ←
• understand and use pseudocode, using the following conditional statements:
IF … THEN … ELSE … ENDIF
CASE … OF … OTHERWISE … ENDCASE
• understand and use pseudocode, using the following loop structures:
FOR … TO … NEXT
REPEAT … UNTIL
WHILE … DO … ENDWHILE
• understand and use pseudocode, using the following commands and statements:
INPUT and OUTPUT (e.g. READ and PRINT)
totalling (e.g. Sum ← Sum + Number)
counting (e.g. Count ← Count + 1)
(Candidates are advised to try out solutions to a variety of different problems on a computer using a language of their choice; no particular programming language will be assumed in this syllabus.)

Introduction to Algorithm

An **algorithm** is a sequence of steps for solving a problem.

In general, an 'algorithm' is the name given to a defined set of steps used to complete a task.

For instance you could define an algorithm to make a cup of tea. You start by filling the kettle, and then place a tea bag in the cup and so on.

In computer terms, an algorithm describes the set of steps needed to carry out a software task. This mini-web takes you through the topic of algorithm

The concept of a program

A program is a sequence of instructions or programming language statements written to make a computer perform certain tasks.

Well-structured programs require a programming language to support the following program constructs:

- sequence
- selection
- iteration

A computer's processor can only run a computer program in the form of a file of machine code, which is a sequence of binary codes representing instructions for the processor. The instruction set for a family of processors is the machine language in which machine code is written for that family of processors.

When machine code runs, the processor repeatedly:

- Fetches an instruction from internal memory
- Decodes the instruction
- Executes the instruction.

Pseudocode

Pseudocode uses keywords commonly found in high-level languages and mathematical notation. It describes an algorithm's steps like program statements, without being bound by the strict rules of vocabulary and syntax of any particular language, together with ordinary English.

Arithmetic operations

Standard arithmetic operator symbols are used:

- + Addition
- - Subtraction
- * Multiplication
- / Division

Care should be taken with the division operation: the resulting value should be of data type REAL, even if the operands are integers.
The integer division operators **MOD** and **DIV** can be used. However, their use should be explained explicitly and not assumed.
INT function is also used in algorithm.

Relational operations

The following symbols are used for relational operators (also known as comparison operators):

- > Greater than
- < Less than
- >= Greater than or equal to
- <= Less than or equal to
- = Equal to
- <> Not equal to

The result of these operations is always of data type BOOLEAN.

Logic operators

The only logic operators (also called relational operators) used are AND, OR and NOT. The operands and results of these operations are always of data type BOOLEAN.

Assignment

Assignment is the process of writing a value into a variable (a named memory location). For example, Count ← 1 can be read as 'Count is assigned the value 1', 'Count is made equal to 1' or 'Count becomes 1'. Another way of indicating assignment is a pseudocode statement such as:

Set Swapped to False

Initialisation:

If an algorithm needs to read the value of a variable *before* it assigns input data or a calculated value to the variable, the algorithm should assign an appropriate initial value to the variable, known as Initialisation.

Input

We indicate input by words such as **INPUT, READ or ENTER,** followed by the name of a variable to which we wish to assign the input value.

Output

We indicate output by words such as **OUTPUT, WRITE or PRINT,** followed by a comma-separated list of expressions.

Totalling

To keep a running total, we can use a variable such as Total or Sum to hold the running total and assignment statements such as:

Total ← Total + Number
ADD Number to Total

Counting

It is sometimes necessary to count how many times something happens.
To count up or increment by 1, we can use statements such as:

Count ← Count + 1
INCREMENT Count by 1

Atomic type names

The following keywords are used to designate atomic data types:

- INTEGER : A whole number (without fractional part)
- REAL : A number capable of containing a fractional part
- CHAR : A single character
- STRING : A sequence of zero or more characters
- BOOLEAN: The logical values TRUE and FALSE
- DATE: A valid calendar date

Variable:

Variable is memory location where a value can be stored. The values stored in a variable are changed during execution.

Variable declarations

It is good practice to declare variables explicitly in pseudocode.
Declarations are made as follows:

DECLARE<identifier> : <data type>

Example

```
DECLARE Surname : STRING
DECLARE FirstName : STRING
DECLARE DateOfBirth : DATE
DECLARE Section : CHAR
DECLARE Counter : INTEGER
DECLARE TotalToPay : REAL
DECLARE GameOver : BOOLEAN
```

Constant:

Constant is memory location where a value can be stored but the stored value remaining same during execution.
It is good practice to use constants if this makes the pseudocode more readable, as an identifier is more meaningful in many cases than a literal. It also makes the pseudocode easier to update if the value of the constant changes.

Constant declaration
Constants are normally declared at the beginning of a piece of pseudocode (unless it is desirable to restrict the scope of the constant).
Constants are declared by stating the identifier and the literal value in the following format:

CONSTANT<identifier> = <value>

Example

CONSTANT HourlyRate = 6.50
CONSTANT DefaultText = "N/A"

Only literals can be used as the value of a constant. A variable, another constant or an expression must never be used.

Declaring arrays
Arrays are considered to be fixed-length structures of elements of identical data type, accessible by consecutive index (subscript) numbers. It is good practice to explicitly state what the lower bound of the array (i.e. the index of the first element) is because this defaults to either 0 or 1 in different systems. Generally, a lower bound of 1 will be used.
Square brackets are used to indicate the array indices.
One- and two-dimensional arrays are declared as follows (where l, l1, l2 are lower bounds and u, u1, u2 are upper bounds):

DECLARE <identifier>: ARRAY[<l>:<u>] OF <data type>
DECLARE <identifier>: ARRAY[<l1>:<u1>,<l2>:<u2>] OF <data type>

Example

DECLARE StudentNames : ARRAY[1:30] OF STRING
DECLARE NoughtsAndCrosses : ARRAY[1:3,1:3] OF CHAR

Using arrays
In the main pseudocode statements, only one index value is used for each dimension in the square brackets.

Example

StudentNames[1] ← "Ali"
NoughtsAndCrosses[2,3] ←'X'
StudentNames[n+1] ←StudentNames[n]

Iteration (Repetition, Loop)

Count-controlled (FOR) loops

Count-controlled loops are written as follows:

```
FOR <identifier>←<value1> TO <value2>
<statements>
NEXT<identifier>
```

The identifier must be a variable of data type INTEGER, and the values should be expressions that evaluate to integers.

It is good practice to repeat the identifier after NEXT.

```
FOR <identifier>←<value1> TO <value2> STEP<increment>
<statements>
NEXT
```

The increment must be an expression that evaluates to an integer. In this case the identifier will be assigned the values from value1 in successive increments of increment until it reaches value2. If it goes past value2, the loop terminates. The increment can be negative.

Example

```
Total = 0
FOR Count = 1 TO 10
INPUT Number
   Total ←Total + Number
NEXT Count
OUTPUT “The grand total is ”, Total
```

Post-condition (REPEAT UNTIL) loops

Post-condition loops are written as follows:

```
REPEAT
<Statements>
UNTIL<condition to stop the loop>
```

The condition must be an expression that evaluates to a Boolean.

The statements in the loop will be executed at least once. The condition is tested after the statements are executed and if it evaluates to TRUE the loop terminates, otherwise the statements are executed again.

Example – REPEAT UNTIL statement

```
REPEAT
   OUTPUT “Please enter the password”
   INPUT Password
UNTIL Password = “Secret”
```

Pre-condition (WHILE) loops

Pre-condition loops are written as follows:

```
WHILE<condition to repeat> DO
    <statements>
ENDWHILE
```

The condition must be an expression that evaluates to a Boolean.
The condition is tested before the statements, and the statements will only be executed if the condition evaluates to TRUE. After the statements have been executed the condition is tested again. The loop terminates when the condition evaluates to FALSE.
The statements will not be executed if, on the first test, the condition evaluates to FALSE.

Example

```
Count ← 0
WHILE Count < 10 DO
  Number ←Number – 9
ENDWHILE
```

Selection:
For selection following statements are used:

- IF
- CASE

IF statements
IF statements may or may not have an ELSE clause.
IF statements without an ELSEclause is written as follows:

```
IF<condition>THEN
    <statements if true>
ENDIF
```

IFstatements with an else clause is written as follows:

```
IF <condition>THEN
    <statements if true>
  ELSE
    <statements if false>
ENDIF
```

Note that the THEN and ELSE clauses are only indented by two spaces. (They are, in a sense, a continuation of the IF statement rather than separate statements).
When IF statements are nested, the nesting should continue the indentation of two spaces. In particular, run-on THENIF and ELSE IF lines should be avoided.

Example

```
IF Number>Largest THEN
Largest←Number
ENDIF
```

CASE statements

CASE statements allow one out of several branches of code to be executed, depending on the value of a variable.
CASE statements are written as follows:

```
CASE OF<identifier>
<value 1> : <statement>
<value 2> : <statement>
  ...
ENDCASE
```

AnOTHERWISEclause can be the last case:

```
CASE OF <identifier>
<value 1> : <statement>
<value 2> : <statement>
  ...
  OTHERWISE<statement>
ENDCASE
```

Example – formatted CASE statement

```
INPUT ItemType
CASE OF ItemType
  1: CD ←CD + 1
  2: DVD ←DVD + 1
  3: Video←Video + 1
  4: Book ←Book + 1
  OTHERWISE : Beep
ENDCASE
```

Writing algorithms in pseudocode

Writing an algorithm in pseudocode is no longer graphical like a program flowchart, but is one step closer to writing program code in a high-level language.

Producing an algorithm for a solution in pseudocode typically includes:

- Declaring variables and constants
- Initialising any variables for total, count, highest and lowest values
- Input values
- Using an appropriate loop structure for repetitions of data entry and/or other processing
- Using conditional statements to select appropriate processing alternatives
- Output the algorithm.

Testing and interpreting pseudocode algorithms

Dry running a pseudocode algorithm with a trace table and test data helps to understand its behaviour and purpose.

```
Line 1 Exponent ← 0
Line 2 REPEAT
Line 3 Result ← 2 ^ Exponent
Line 4 PRINT "2^", Exponent, " = ", Result
Line 5 Exponent ← Exponent + 1
Line 6 UNTIL Result > 100
```

What does this pseudocode algorithm do?

Trace Table

Trace tables are used to dry run of algorithm for testing.

Trace Table has columns for all variables, logical expressions and output.

Exponent	Result	OUTPUT	Description
0			Initialisation of variable
1	1	2^0=1	1st iteration
2	2	2^1=2	2nd iteration
3	4	2^2=4	3rd iteration
4	8	2^3=8	4th iteration
5	16	2^4=16	5th iteration
6	32	2^5=32	6th iteration
7	64	2^6=64	7th iteration
8	128	2^7=128	8th iteration

Example interpretation

The purpose of the algorithm is to print a list of the powers of 2 starting at 20 until it reaches the first one over 100.

Finding and correcting errors in pseudocode algorithms

It is important to be able to identify errors and suggest corrections in a pseudocode algorithm.

Finding and correcting errors in pseudocode algorithm

There are 8 types of errors in pseudo code:

Error 1: Faulty initial or final value of loop counter

IF Count is initialized with 0 then Count< 'number of iteration' should be used in WHILE loop.
IF Count is initialized with 1 then Count<= 'number of iteration' should be used in WHILE loop.

```
A computer program is required which inputs 10 numbers,
finally outputs the answer (the product). The following algorithm
1    count = 0
2    while count <= 10 do
```

```
1.   SET X = 1
2.   REPEAT
3.        X = X + 2
4.        Print X
5.   UNTIL X = 10
```

Error 2: Missing or Faulty initialization of a variable:

A variable must be initialized if it used in calculation without INPUT.
Total is initialized with 0, Product with 1, Highest with lowest possible value and Lowest with highest possible value.

```
10  total = 1
```

```
10  highest = 0
20  lowest = 0
```

```
1    count = 0
2    product = 0
```

```
10  count = 0
20  REPEAT
30      INPUT n
40      sum = sum + n
```

Error 3: Increment in loop Counter in FOR…TO…NEXT loop.

FOR…TO…NEXT loop doesn't need increment in loop counter.

```
20  FOR x = 1 TO 500
30      IF number < 10 THEN total = total + 1
40      k = x / number
50      x = x + 1
```

```
30  for count = 1 to 100
40      input number
50      if number > highest then highest = number
60      if number < lowest then  lowest = number
70      count = count + 1
80  next count
```

Error 4: Missing increment in loop Counter in REPEAT…UNTIL or WHILE…DO…ENDWHILE loop.

REPEAT…UNTIL loop and WHILE..DO..ENDWHILE loop needs increment in loop counter.

```
1 c= 0
2 h=0
3 REPEAT
4       READ x
5       IF x>h THEN h=x
6  UNTIL c>=20
7 OUTPUT h
```

Error 5: Misplacing statement inside or outside of loop:

If Final Output like greatest value or average is required it should be after loop.
IF running output is required it should be inside loop.

```
for count = 1 to 20 do
    input number
    if number < 0 then negative = negative + 1
    if number > 0 then positive = positive + 1
    print negative, positive
next count
```

Error 6: Missing ending keywords.

REPEAT…UNTIL or
WHILE…DO…ENDWHILE
FOR…TO…NEXT
IF…THEN…ENDIF
CASE ….OF…OTHERWISE….ENDCASE

```
1.  SET X = 1
2.  REPEAT
3.        X = X + 2
4.  Print X
```

```
set Total_1 to zero
set Counter to one
while Counter < eight
    Counter = Counter + 1
    input Number
    if Number > zero then Total_1 = Total_1 + Number
output Total_1
```

Error 7: Assignment Error.

Values or vales of variable at right side should be assigned to variables and constants at left side.

```
    Number ← 58
    Highest ← Number
30  for count = 1 to 100
40      input number
50      if number > highest then number = highest
60      if number < lowest then number = lowest
70      count = count + 1
80  next count
```

Error 8: Operator Error.
A common error in pseudocode is an improper operator.

IF number < Highest THEN Highest ←Number

```
30  for count = 1 to 100
40        input number
50        if number < highest then highest = number
60        if number > lowest then  lowest = number
70        count = count + 1
80  next count
```

Examination Questions

Q1) Winter 2014 P13

The following pseudocode algorithm should:
• input up to 20 numbers
• stop if the sum of the input numbers **exceeds** 50
• output the final sum

```
10 count = 0
20 REPEAT
30 INPUT n
40 n + sum = sum
50 IF sum = 50 THEN count = 20
60 count = count + 1
70 UNTIL count = 20
80 OUTPUT n
```

There are **five** errors in this algorithm.
Locate these errors and suggest a correction.

error 1 .....
correction .....
.....
error 2 .....
correction .....
.....
error 3 .....
correction .....
.....
error 4 .....
correction .....
.....
error 5 .....
correction .....
.....[5]

Q2) Summer 2005

The following algorithm contains an error.

1. SET X = 1
2. REPEAT
3. X = X + 2
4. Print X
5. UNTIL X = 10

(a) Trace the algorithm and explain what the error is.

.....
..... [2]

Q3) Winter 2006

A computer program is required which inputs 10 numbers, multiplies them together and finally outputs the answer (the product). The following algorithm has been written to do this.

```
1       count  = 0
2       product  = 0
3       while  count<= 10 do
4       input number
5       product  = product  * number
6       count  = count  + 1
7       print product
8       endwhile
```

(a) There are three errors in the algorithm. Locate and describe these errors.

A while do loop has been used in the algorithm. State another type of loop that could have been used.

Summer & Winter 2007, Summer & Winter 2008, Summer and winter 2009

No questions on error detection

Q4) Summer 2010

A golf course charges $10 for each game of two people. Each additional person incurs a further charge of $2 per game. If they book two or more games in advance, they get a 10% discount on the total charge.The following program has been written in pseudocode to calculate the charges for a game.

```
1       extracost = 0
2       input numberpeople, numbergames
3       charge = 10 * numbergames
4       extrapeople = numberpeople – 2
5       if numberpeople< 2 then extracost = 2 * extrapeople * numbergames
6       charge = extracost
7       if numbergames> 1 then charge = charge * 0.1
8       print charge
```

There are three errors in the program. Locate these errors and suggest a correct piece of coding.

Error 1 .....

Correction 1 .....
Error 2 .....
Correction 2 .....
Error 3 .....
Correction 3 ..... [6]

Q5) Winter 2010

The following algorithm inputs 20 numbers and outputs how many numbers were positive (> 0) and how many numbers were negative (< 0).

```
1       negative = 1
2       positive  = 1
3       for count  = 1 to 20 do
4       input number
5       if number < 0 then negative = negative + 1
6       if number > 0 then positive  = positive  + 1
7       count  = count  + 1
8       print negative, positive
9       next count
```

There are three different errors in this algorithm.
Locate each error and give the reason why you think it is an error.
Error 1: ……..
Correction 1: ……..
Error 2: ……..
Correction 2: ……..
Error 3: ……..
Correction 3: ..... [6]

Q6) Summer 2011

Read the following section of code that inputs twenty (20) numbers and then outputs the largest number input.

```
1 h = 0
2 c = 0
3 REPEAT
4 READ x
5 IF x > h THEN x = h
6 c = c + 1
7 PRINT h
8 UNTIL c < 20
```

There are THREE errors in this code.
Locate these errors and suggest a corrected piece of code.
Error 1: ……..
Correction 1: ……..
Error 2: ……..
Correction 2: ……..
Error 3: ……..
Correction 3: ..... [6]

Q7) Winter 2013

A piece of pseudocode was written to input 1000 positive numbers and then output the highest and lowest numbers.

```
10 highest = 0
20 lowest = 0
30 for count = 1 to 100
40      input number
50      if number > highest then number = highest
60      if number < lowest then number = lowest
70      count = count + 1
80 next count
90 print highest, lowest
```

There are errors in the code.

Locate these errors and suggest a correction.

error 1 .....

correction: .....

.....

error 2: .....

correction: .....

.....

error 3: .....

correction: .....

.....

error 4: .....

correction .....

.....

Q8) Winter 2014 P12

The following section of a pseudocode algorithm should:

- input 500 numbers
- generate a ratio called **k**
- output each value of **k**
- output how many numbers were larger than 10

```
10 total = 1
20 FOR x = 1 TO 500
30      IF number < 10 THEN total = total + 1
40      k = x / number
50      x = x + 1
60      OUTPUT k
70 NEXT x
80 OUTPUT x
```

(a) There are **five** errors in the above code.

Locate these errors and suggest a correction.

error 1 .....

correction: .....
.....
error 2: .....
correction: .....
.....
error 3: .....
correction: .....
.....
error 4: .....
correction .....
.....
error 5: .....
correction .....
.....

(b) The corrected algorithm was converted to a computer program and run. However, after several numbers were input, the program stopped and an error message was generated, showing that there was a further error at line 40 (k = x / number).
State what could cause this error to occur.

.....
.....

Suggest a change to line 40 to overcome this problem.

.....
.....
.....[2]

Marking Scheme (error detection)

Q1)1 mark for each error located with corresponding correction (description or corrected pseudocodeacceptable)
error: line 10: sum not initialised
correction: sum = 0
error: line 40: incorrect formula for sum
correction: sum = sum + n
error: line 50: incorrect IF statement
correction: IF sum > 50 THEN ……
error: lines 50 and 60: value of count causes a problem e.g. loop never ending
correction: either count = 19 on line 50or count = count + 1 between lines 30 and 40or any other correct solution
error: line 80: output of n is incorrect
correction output sum or print sum[5]

Q2) (a) Award 1 mark each for trace and reason:
trace – 3,5,7,9,11……
reason – x is odd/loop does not terminate/goes on forever [2]

Q3) (a) error 1: product = 0 on line 2
should use product = 1
error 2: loop control, count <= 10 on line 3
should use count < 10 or alternatively alter count value on line 1 to count = 1
error 3: print value of product inside loop on line 7
output should come after the endwhile statement [3]
(b) Accept either of the following loop controls:
repeat for count = 1 to 10ORuntil count = 10 next count(accept repeatuntil count > 11if line 1 changed to count = 1) [1]

Q4) 1 mark for each error identified + 1 mark for each suggested correction
– error: line 5: numberpeople < 2 is incorrect
correction: numberpeople > 2
– error: line 6: the formula/charge = extracost is incorrect
correction: charge = extracost + charge
*– error: line 7: discount calculation/charge = charge * 0.1 is incorrect,*
*correction: charge = charge * 0.9*

Q5) mark for each error and 1 mark for reason why it is an error
– line 1/negative=1 and/or line 2/positive=1
– negative and/or positive should be set to zero
– line 7/count=count+1
– don't need a count within a for …. to next loop
– replace loop with a repeat...until loop
– line 8/print negative, positive or line 9/next count
– outputs should come after the next count statement

Q6) (a) 1 mark for each error identified + suggested correction
line 5: this should read if x > h then h = x
line 7: print h should come after the end of the repeat loop
line 8: this should read until c = 20 or until c >= 20 or until c > 19

Q8) (a) 1 mark for each error and suggested correction (accept description or example of corrected
pseudocode).
error: line 10: total = 1
correction: totals should be set to zero; total = 0
error: line 30: … number < 10 …
correction: check should be made if number > 10; … number > 10 …
error: no input inside loop
correction: input number
error: line 50: x = x + 1
correction: for … to loops don't need a counter; remove line 50 altogether
error: line 80: output x
correction: output should be total value; output total [5]
(b) division by zero error (or similar description of error produced when dividing by 0) add an error trap after input of number
e.g. 40 if number = 0 then k = 0 else k = x/number [2]

Finding output from pseudocode

Q1) Winter 2001

This algorithm grades candidates on marks out of ten.

```
1 input a Mark
2    case Mark of
3         0, 1, 2, 3 : Grade = Fail
4         4, 5 : Grade = Pass
5         6, 7 : Grade = Merit
6         8, 9, 10 : Grade = Distinction
7         otherwise Mark = -1
8 endcase
9 if Mark = -1 then
10        print 'Not a valid mark'
11 else output Grade, 'Grade'
```

(a) Dry run the algorithm for each of the following data and complete the table. [3]

INPUT	OUTPUT
0	
5	
99	

(b) Write down two instructions which could be inserted between lines 1 and 2 to allow the algorithm to deal with marks out of 100. [2]

Q2) Specimen 2015

Jatinder uses Internet banking. This pseudocode checks her PIN.

```
c ← 0
INPUT PIN
x ← PIN
REPEAT
        x ← x/10
        c ← c + 1
UNTIL x < 1
IF c <> 5THEN
        PRINT "error in PIN entered"
ELSE
        PRINT "PIN OK"
ENDIF
```

(a) What value of c and what message would be output if the following PINs were entered?

5 1 0 2 0 Value of c:.....

Message:.....

5 1 2 0 Value of c:.....

Message: .....[2]

(b) What type of validation check is being carried out here?

.....[1]

Q3) Winter 2002

Read this algorithm.

```
set Total_1 to zero
set Total_2 to zero
set Counter to one
while Counter < eight
        Counter = Counter + 1
        input Number
        if Number > zero then Total_1 = Total_1 + Number
        if Number < zero then Total_2 = Total_2 + Number
endwhile
output Total_1
output Total_2
```

(a) Write down the output if the following set of numbers are input. 4, 1, -3, 2, -5, 0, 6

..... [2]

(b) Modify the algorithm so that it will accept any number of numbers, the input is terminated by a rogue value and the output is the Total of all the numbers input except the rogue value.

..... [4]

Q4)Summer 2003

Read this algorithm.

```
input A, B
if A > B then
        T = A
        A = B
        B = T
endif
output A, B
```

(a) Write down the output if the following two numbers are input:

41, 38

.....[1]

(b) Explain the purpose of the variable T ..... [1]

(c) Explain why an algorithm is written as a subroutine (procedure) and stored in a program library. .....[2]

Q5) Winter 2003

The following algorithm inputs air speeds (which must be in multiples of 100) and outputs asuitable message.

```
1 input a speed
2 whole = speed/100
3 case whole of
4       0,1,2 : result = slow
5       3, 4, 5, 6 : result = normal
8       7, 8, 9 : result = high
7       otherwise whole = -1
8 endcase
9 if whole = -1 then
10      output “abnormal reading”
11 else output result, “speed”
```

Dry run the above algorithm for the following Input data and complete the Output column in the table: [3]

Input	Output
150	
400	
800	

State what would be happen if line 2 had been missed out of the algorithm?

.....

..... [2]

Q6) (b) Write an algorithm which uses a While..Do..Endwhile loop and outputs the numbers 2, 4, 6 and 8[3]

Marking Scheme

Q1)a

Input	*Output*
0	*Fail Grade*
5	*Pass Grade*
99	*Not a valid mark*

b) For example(1 mark per line)

mark = mark/10

mark= INT(Mark)

or mark = mark DIV 10 is worth 2 marks on its own

Writing Algorithm

Producing algorithms in pseudocode

Writing an algorithm in pseudocode is no longer graphical like a program flowchart, but is one step closer to writing program code in a high-level language.

Producing an algorithm for a solution in pseudocode typically includes:

- Declaring variables using data types.
- Initialisingany variables for totalling, highest, Lowest and counting
  - Count← 0 [Count starts with 0, like Count_Integer, Count_Above]
  - Highest← 0 [The lowest possible value]
  - Lowest ← 999 [The highest possible value]
  - Total ← 0

- UsingREPEAT…UNTIL or WHILE…DO…ENDWHILE for input validation
- Using FOR…TO…NEXT when number of repetitions are given like number of students, houses, days
- Using conditional statements to select appropriate processing alternatives
- IF…THEN…ELSE…ENDIF statements for adjusting the values of maximum and minimum variables cannot be nested.
  - Highest ←0
    INPUT Number
    IF Number > Highest THEN Highest ← Number

  - Lowest ←0
    INPUT Number
    IF Number<Lowest THEN Highest ← Number

Input

We indicate input by words such as **INPUT, READ or ENTER,** followed by the name of a variable to which we wish to assign the input value.

Output

We indicate output by words such as **OUTPUT, WRITE or PRINT,** followed by a comma-separated list of expressions.

Totalling

To keep a running total, we can use a variable such as Total or Sum to hold the running total and assignment statements such as:

Total ← Total + Number [ADD Number to Total]

Counting

It is sometimes necessary to count how many times something happens.

To count up or increment by 1, we can use statements such as:

Count ← Count + 1 [INCREMENT Count by 1]

Q1) Using pseudo code or otherwise, write an algorithm which will output following sequence of numbers:

a. 1, 2, 3, 4, 5, 6,7,8,9,10
b. 1,3,5,7,9,11,13
c. 2,4,6,8,10,12,14,16,18,20
d. 10,9,8,7,6,5,4,3,2,1
e. 2,6,10,14,18,22,26,30

Q2) Using pseudo code or otherwise, write an algorithm which will output following sequence of numbers:

1, 2, 3, 4, 5	1,1,1,1,1	1	* * * * *
1, 2, 3, 4, 5	2,2,2,2,2	1,2	* * * *
1, 2, 3, 4, 5	3,3,3,3,3	1,2,3	* * *
1, 2, 3, 4, 5	4,4,4,4,4	1,2,3,4	* *
1, 2, 3, 4, 5	5,5,5,5,5	1,2,3,4,5	*

Q3) Winter 2003

(a) Write an algorithm, using pseudocode or otherwise, which;

- inputs 50 numbers
- checks whether each number is in the range 1000 to 9999
- outputs how many of the input numbers were out of range
- outputs the percentage of input numbers which were out of range.

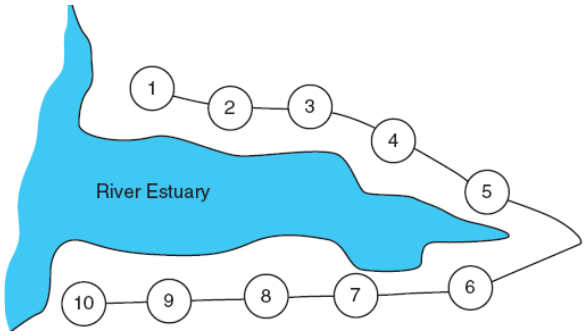
Q4) Summer 2004

(b) Using pseudocode, or otherwise, write an algorithm that will input the hourly temperatures for one day in Centigrade and print out in Fahrenheit

- the maximum temperature
- the minimum temperature
- the average temperature for that day.

Q5) Winter 2004

The following diagram shows a rail network.



The rail network consists of 10 stations. The fare between each station is $2. There is a 10% discount when 3 or more passengers travel together. Tickets can be purchased at any station using automated terminals.

Using pseudocode, or otherwise, write an algorithm for the automated terminals to:

• input the starting station number, the destination station number and the number of passengers

• calculate the total fare and output the amount to be paid

• calculate the change (if any)

• issue the rail ticket(s) and change

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q6) Summer 2005

Using pseudocode or otherwise, write an algorithm that will input 25 marks and output the number of DISTINCTION, MERIT, PASS or FAIL grades.

A mark greater than 69 will get a DISTINCTION, a mark between 69 and 60 (inclusive) will get a MERIT and a mark between 59 and 50 (inclusive) will get a PASS.

Q 7) Winter 2005

A school uses a computer to store student marks obtained in an end of term mathematics exam. There are 150 students doing the exam and the maximum mark is 100.

Write an algorithm, using pseudocode or otherwise, which

- inputs the marks for all students
- checks if each mark is in the correct range and, if not, the mark is re-input
- outputs the smallest mark
- outputs the highest mark
- outputs the average mark for the exam.

Q8) Summer 2006

(a) A formula for calculating the body mass index (BMI) is:

$$BMI = \frac{\text{weight in kilograms}}{(\text{height in metres}) \times (\text{height in metres})}$$

Calculate the BMI for a person whose weight is 80kg and height is 2 meters.

(b) Using pseudocode or otherwise, write an algorithm that will input the ID, weight (kg) and height (m) of 30 students, calculate their body mass index (BMI) and output their ID, BMI and a comment as follows:

A BMI greater than 25 will get the comment 'OVER WEIGHT', a BMI between 25 and 19 (inclusive) will get 'NORMAL' and a BMI less than 19 will get 'UNDER WEIGHT'.

Q9) Winter 2006

Temperatures (°C) are being collected in an experiment every hour over a 200 hour period.

Write an algorithm, using pseudocode or otherwise, which inputs each temperature and outputs

- how many of the temperatures were above 20°C
- how many of the temperatures were below 10°C
- the lowest temperature that was input

Q10) Summer 2007

A company has 5000 CDs, DVDs, videos and books in stock. Each item has a unique 5-digit code with the first digit identifying the type of item, i.e.

1 = CD
2 = DVD
3 = video
4 = book

For example, for the code 15642 the 1 identifies that it is a CD, and for the code 30055 the 3 identifies that it is a video.

Write an algorithm, using pseudocode or otherwise, that

- Inputs the codes for all 5000 items
- Validates the input code
- Calculates how many CDs, DVDs, videos and books are in stock
- Outputs the four totals.

Q11) Winter 2007

(a) Fuel economy for a car is found using the formula:

$$\text{Fuel Economy} = \frac{\text{Distance Travelled (km)}}{\text{Fuel Used (litres)}}$$

What would be the Fuel Economy of a car travelling 40 km on 10 litres of fuel?

(b) The Fuel Economy for 1000 cars is to be calculated using the formula in Question 16(a).

Write an algorithm, using pseudocode or otherwise, which inputs the Distance Travelled (km) and the Fuel Used (litres) for 1000 cars. The Fuel Economy for each car is then calculated and the following outputs produced:

- Fuel Economy for each car
- average (mean) Fuel Economy for all of the cars input
- the best Fuel Economy (i.e. highest value)
- the worst Fuel Economy (i.e. lowest value)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Q12) Summer 2008

Customers can withdraw cash from an Automatic Teller Machine (ATM).

- withdrawal is refused if amount entered > current balance
- withdrawal is refused if amount entered > daily limit
- if current balance < $100, then a charge of 2% is made
- if current balance $100, no charge is made

Write an algorithm which inputs a request for a sum of money, decides if a withdrawal can be made and calculates any charges. Appropriate output messages should be included.

Q13) Winter 2008

The manufacturing cost of producing an item depends on its complexity. A company manufactures three different types of item, with costs based on the following calculations:

Item type 1: item cost = parts cost * 1.5
Item type 2: item cost = parts cost * 2.5
Item type 3: item cost = parts cost * 5.0

The company makes 1000 items per day.

Write an algorithm, using pseudocode, flowchart or otherwise, which

• inputs the item type and parts cost of each item
• outputs the item cost for each item
• calculates and outputs the average (mean) item cost per day (based on 1000 items being made).

Q14) Summer 2009

A small airport handles 400 flights per day from three airlines:
FASTAIR (code FA)
SWIFTJET (code SJ)
KNIGHTAIR (code KA)
Each flight is identified by the airline code and 3 digits. For example FA 156.
Write an algorithm, using pseudocode or otherwise, which monitors the 400 flights into and out of the airport each day. The following inputs, processing and outputs are all part of the monitoring process:

- input flight identification
- calculate number of flights per day for each of the three airlines
- output the percentage of the total flights per day by each airline
- any validation checks must be included

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Q15) Winter 2009

(a) A car's speed is measured between points A and B, which are 200 km apart.

What is the final speed of a car if it takes 2 hours to get from A to B?

(b) Write an algorithm, using pseudocode or otherwise, which inputs the times for 500 cars, calculates the final speed of each car using the formula in part (a), and then outputs:

- the final speed for ALL 500 cars
- the slowest (lowest) final speed
- the fastest (highest) final speed
- the average final speed for all the cars.

Q16) Summer 2010

A group of students were monitoring the temperature every day over a one-year period. Readings were taken ten times every day (you may assume a year contains 365 days). Write an algorithm, using pseudocode or flowchart, which

- inputs all the temperatures (ten per day)
- outputs the highest temperature taken over the year
- outputs the lowest temperature taken over the year
- outputs the average temperature per day
- outputs the average temperature for the whole year

(a) Write an algorithm, using pseudocode or a flowchart, which:
_ inputs 50 numbers
_ outputs how many of the numbers were > 100
(b) Write an algorithm, using pseudocode or a flowchart, which:
_ inputs 100 numbers
_ finds the average of the input numbers
_ outputs the average

Q17) Winter 2010

A school is doing a check on the heights and weights of all its students. The school has 1000 students.

Write an algorithm, using pseudocode or a flowchart, which

- inputs the height and weight of all 1000 students
- outputs the average (mean) height and weight
- includes any necessary error traps for the input of height and weight

17 (a) Write an algorithm, using pseudocode or a flowchart, which
_ inputs a set of positive numbers (which end with -1)
_ outputs the average (mean) value of the input numbers
_ outputs the value of the largest (highest) number input
(b) Write an algorithm, using pseudocode or a flowchart, which
_ inputs a whole number (which is > 0)
_ calculates the number of digits in the number
_ outputs the number of digits and the original number
(E.g. 147 would give an output of 3, 147)

Q18) Specimen 2011

The exchange rate between the US Dollar (US$) and the Brazilian Real (R$) changes every day.

Write an algorithm, using pseudocode or otherwise, which inputs the exchange rate for every day over a 10 year period (assume that each year = 365 days) and then outputs the following:

• The average (mean) exchange rate
• The best (highest) exchange rate
• The worst (lowest) exchange rate
• The number of occasions when the exchange rate was above 2.0

Q19) Summer 2011

Daniel lives in Italy and travels to Mexico, India and New Zealand. The times differences are:

Country	Hours	Minutes
Mexico	-7	0
India	+4	+30
New Zealand	+11	0

Thus, if it is 10:15 in Italy it will be 14:45 in India.

(a) Write an algorithm, using pseudocode or otherwise, which:

- Inputs the name of the country
- Inputs the time in Italy in hours (H) and minutes (M)
- Calculates the time in the country input using the data from the table
- Outputs the country and the time in hours and minutes

A school has 1800 students. The start date and leaving date for each student is stored on file. Dates are in the format YYMMDD (e.g. a student starting on 10th September 2007 and leaving on 4th August 2012 has the data 070910 and 120804 on file).

(a) Write an algorithm, using pseudocode or otherwise, which

- inputs Student ID for all 1800 students
- inputs the start date and leaving date for each student
- carries out a check to ensure the second date is later
- if error, increments error counter
- outputs the number of errors

Q20) Winter 2011

(a) Write an algorithm, using pseudocode or flowchart only, which:
• inputs three numbers
• outputs the largest of the three numbers
(b) Write an algorithm, using pseudocode or flowchart only, which:
• inputs 1000 numbers
• outputs how many of these numbers were whole numbers (integers)
(You may use INT(X) in your answer e.g. Y = INT(3.8) gives the value Y = 3)

The weather conditions in a town are being monitored over a year (365 days). The values recorded per day are weather type and temperature (e.g. CLOUDY, 25).

Write an algorithm, using pseudocode or flowchart only, which:

- inputs the weather type and temperature for each day
- outputs the number of days that were CLOUDY, RAINING, SUNNY or FOGGY
- outputs the highest recorded temperature for the year
- outputs the lowest recorded temperature for the year

Q21) Summer 2012

Write an algorithm, using pseudocode or a program flowchart only, which:

- inputs the population and land area for 500 countries,
- calculates the population density (i.e. population/land area) for every country,
- outputs the largest and smallest population density,
- outputs the average population for all 500 countries.

An estate agent advertises houses for sale. The customer enquiries for a 7-day working week are entered weekly into a computer.

Write an algorithm, using pseudocode or a program flowchart only, which:

- inputs the number of customer enquiries each day,
- inputs the house price each customer enquires about,
- outputs how many customers enquired each day about houses costing less than $100 000,
- outputs the percentage of all enquiries made during the week about houses costing more than $500 000.

Q22) Winter 2012

Write an algorithm, using pseudocode or a program flowchart only, that:
• inputs a series of positive numbers (-1 is used to terminate the input),
• outputs how many numbers were less than 1000 and
• outputs how many numbers were greater than 1000.

16 A small café sells five types of item: bun 0.50 dollars
coffee 1.20 dollars
cake 1.50 dollars
sandwich 2.10 dollars
dessert 4.00 dollars

Write an algorithm, using pseudocode or a program flowchart only, which
• inputs every item sold during the day,
• uses an item called “end” to finish the day’s input,
• adds up the daily amount taken for each type of item,
• outputs the total takings (for all items added together) at the end of the day,
• outputs the type of item that had the highest takings at the end of the day.

Q23) Summer 2013

A small shop uses barcodes which represent 5 digits. The last digit is used as a check digit.

For example:

a b c d e

0 1 2 3 4

The check digit (e) is found by:

• multiplying the first and third digits (i.e. a and c) by 3
• multiplying the second and fourth digits (i.e. b and d) by 2
• adding these four results together to give a total
• dividing this total by 10
• remainder is check digit (e)

Write an algorithm, using pseudocode or flowchart only, which

• inputs 100 five-digit barcodes in the form a, b, c, d, e
• re-calculates the check digit for each number and checks whether the input check digit (e) is correct
• outputs the number of barcodes which were entered correctly

A country has four mobile phone network operators. Each mobile phone number has eight digits. The first three digits identify the network operator:

444 Yodafone
555 N2 network
666 Kofee mobile
777 Satsuma mobile

Write an algorithm, using pseudocode or flowchart only, which reads 50 000 eight-digit mobile phone calls made during the day and outputs the number of calls made on each of the four networks.

Q24) Winter 2013

(a) A greenhouse is being monitored by a computer using 2 sensors. SENSOR1 measures the temperature and SENSOR2 measures oxygen levels.
If the temperature exceeds 45°C or oxygen levels fall below 0.19, then an error message is output by the computer.
Write an algorithm, using pseudocode or flowchart only, which
• inputs both sensor readings
• checks the sensor input values and outputs a warning message if either are out of range
• continues monitoring until the <ESCAPE> key is pressed
(You may assume that READ SENSORn will take a reading from SENSORn and that READ KEY inputs a key press from the keyboard).

5000 numbers are being input which should have either 1 digit (e.g. 5), 2 digits (e.g. 36), 3 digits (e.g. 149) or 4 digits (e.g. 8567).

Write an algorithm, using pseudocode or flowchart only, which

- inputs 5000 numbers
- outputs how many numbers had 1 digit, 2 digits, 3 digits and 4 digits
- outputs the % of numbers input which were outside the range

Q25) Summer 2014 P11

A survey is being carried out which involves reading and recording sound levels near a busy road junction. Once all the data are collected, they are input manually into a computer. A sound level of 0 decibels (0 dB) is input to indicate the end of the data.
Write an algorithm, using pseudocode or a flowchart, which:

- inputs all the sound levels
- after a sound level of 0 is input, outputs the following:
  - average sound level
  - highest recorded sound level.

Q26) Summer 2014 P12

A school has 1500 students. It is conducting a survey on their music preferences. Each student uses a computer and inputs their name and then chooses one of 5 options:

- rock (input value 1)
- soul (input value 2)
- pop (input value 3)
- jazz (input value 4)
- classical (input value 5)

Write an algorithm, using pseudocode or a flowchart, which:

- inputs the choice of all 1500 students (values 1 to 5)
- outputs all the names of the students who chose classical music
- outputs the percentage who chose each option.

Q27) Winter 2014 P12

A school has 3000 students sitting final examinations.

Each student sits eight examinations.

Write an algorithm, using pseudocode **or**a flowchart, which:

• inputs the marks for **all** 8 examinations for **each** student

• outputs for each student the average mark for their 8 examinations

• outputs the highest mark overall

Q28) Winter 2014 P13

A customer wants to compare prices of 1000 items sold in two supermarkets (price1 and price2).

Write an algorithm, using pseudocode**or**a flowchart, which:

• inputs the two prices for all 1000 items

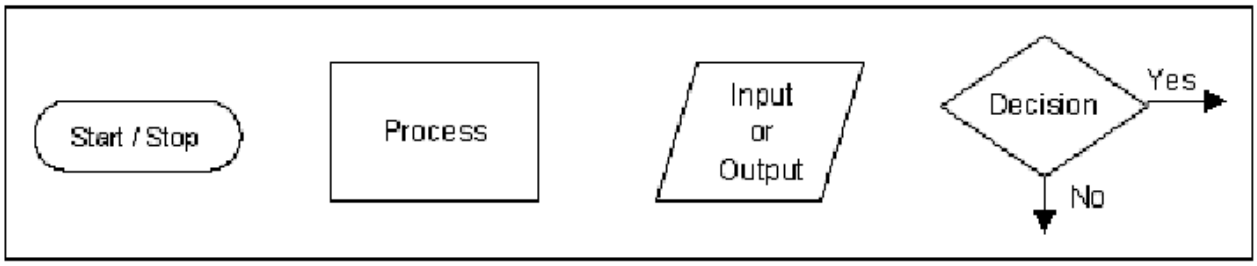
• outputs how many items were more expensive in supermarket 1

• outputs how many items were more expensive in supermarket 2

• outputs the largest price difference

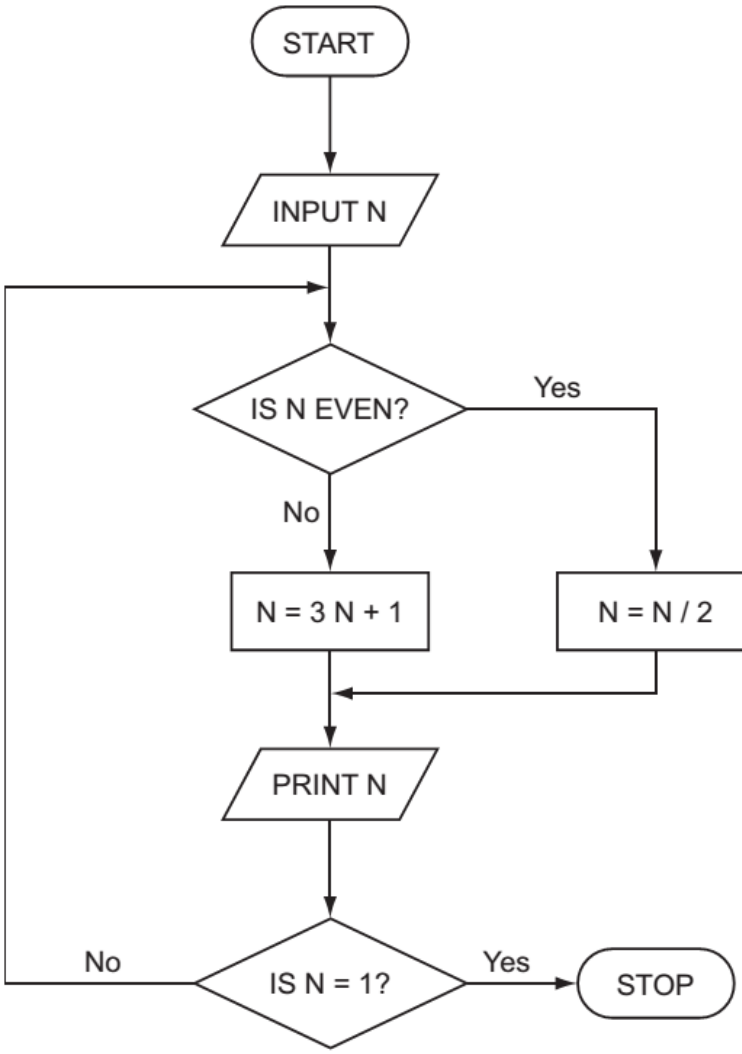
2.1.2 Flowchart

A flowchart is another way of breaking down a program in the form of a diagram.
The following are recognised flowchart symbols:



Finding Output from flowchart

Q1: Summer 2006



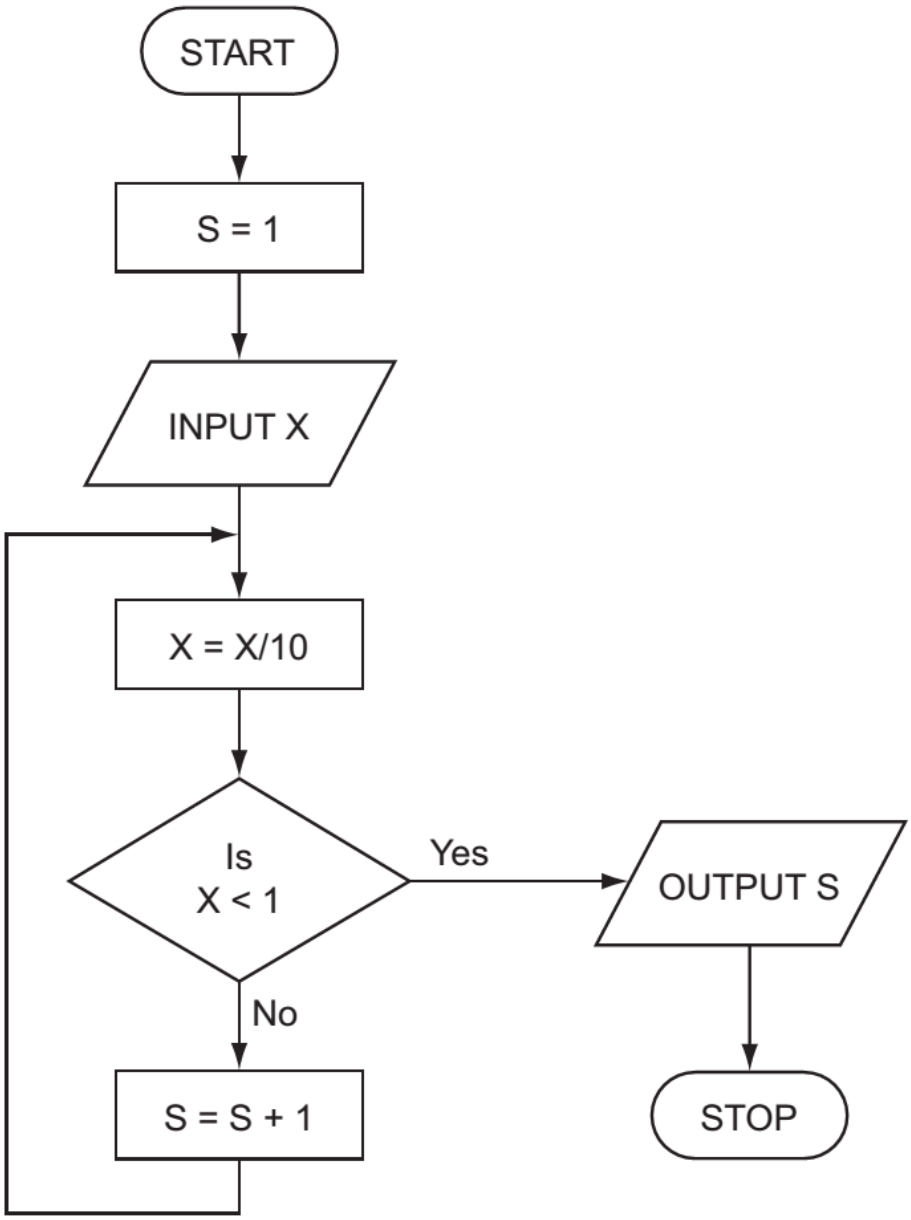
Trace the flow chart using the numbers 2 and 3. Write down each of the values of N in the orderthat they are printed out.

(a) 2 .....[1]

(b) 3 .....[2]

Q2: Summer 2007

Study the following flowchart very carefully.



(a) Complete the following table showing the expected output from the flowchart for the three sets of input data: [3]

INPUT X	OUTPUT S
48	
9170	
- 800	

(b) Input data needs to go through a validation process.

(i) Explain the term validation.

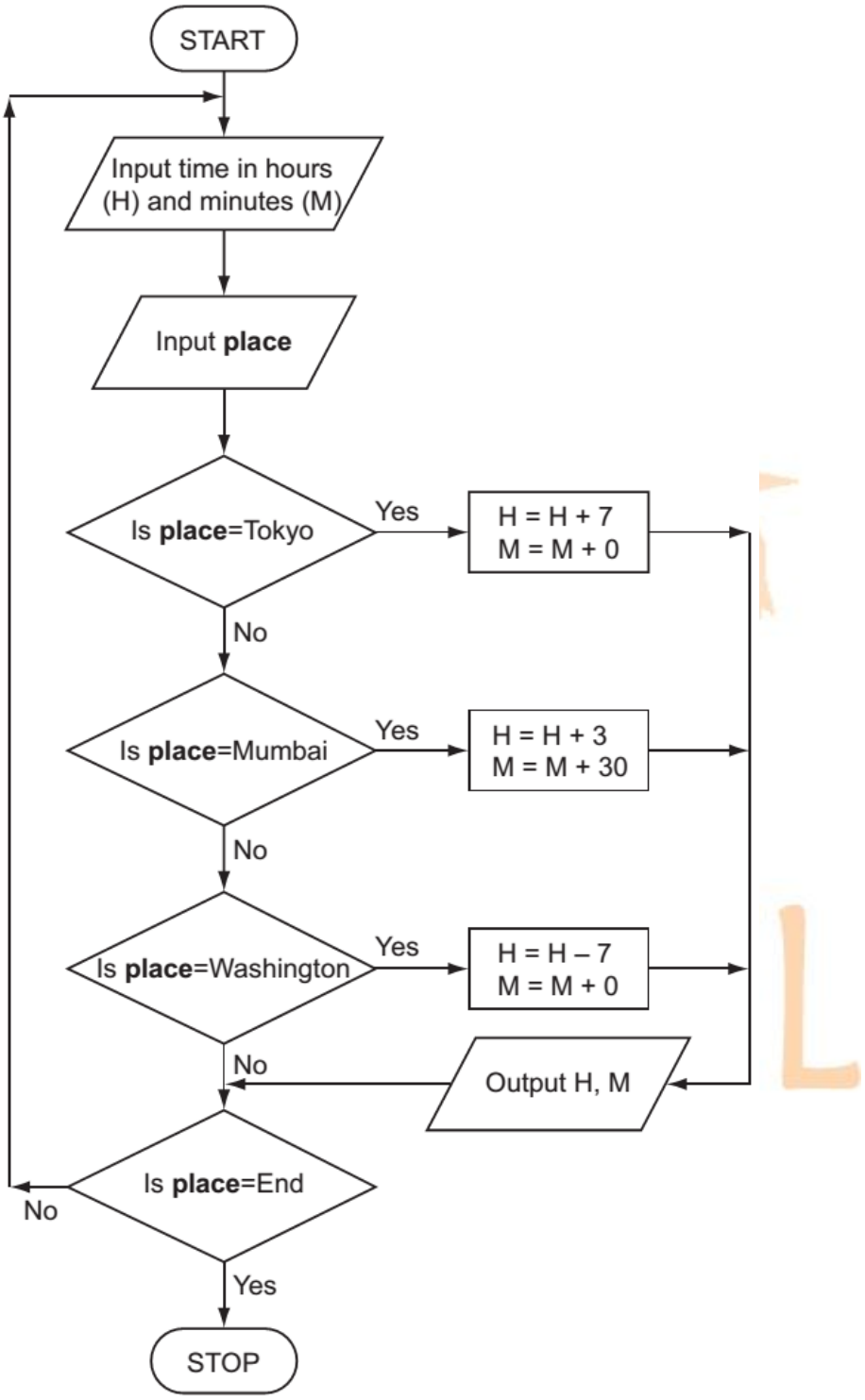
……

(c) (ii) Describe one type of validation check

…… [2]

Q3:Winter 2007

Majid lives in Cairo but often travels to Tokyo, Mumbai and Washington. A flow chart hasbeen written so he can work out the local time in these three places.



(a) What output would be produced from the following input? [2]

Input			Output	
place	hours (H)	minutes (M)	H	M
Tokyo	11	15		
Mumbai	15	10		

(b) What problem would occur if place = Mumbai and H = 15 and M = 30?

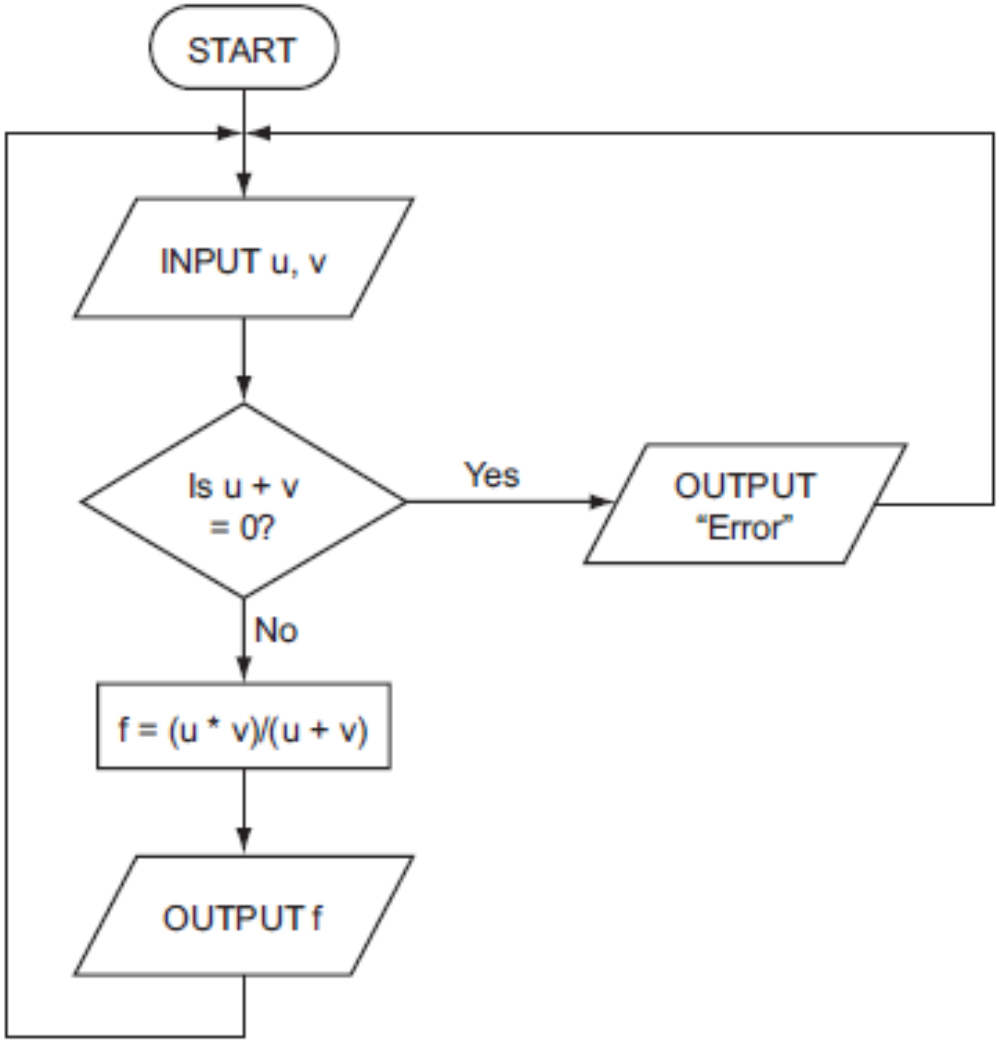
…… [1]

(c) What problem would occur if place = Washington and H = 4 and M = 0?

…… [1]

Q4:Summer 2008

The following flowchart inputs two numbers, carries out a calculation and then outputs the result.



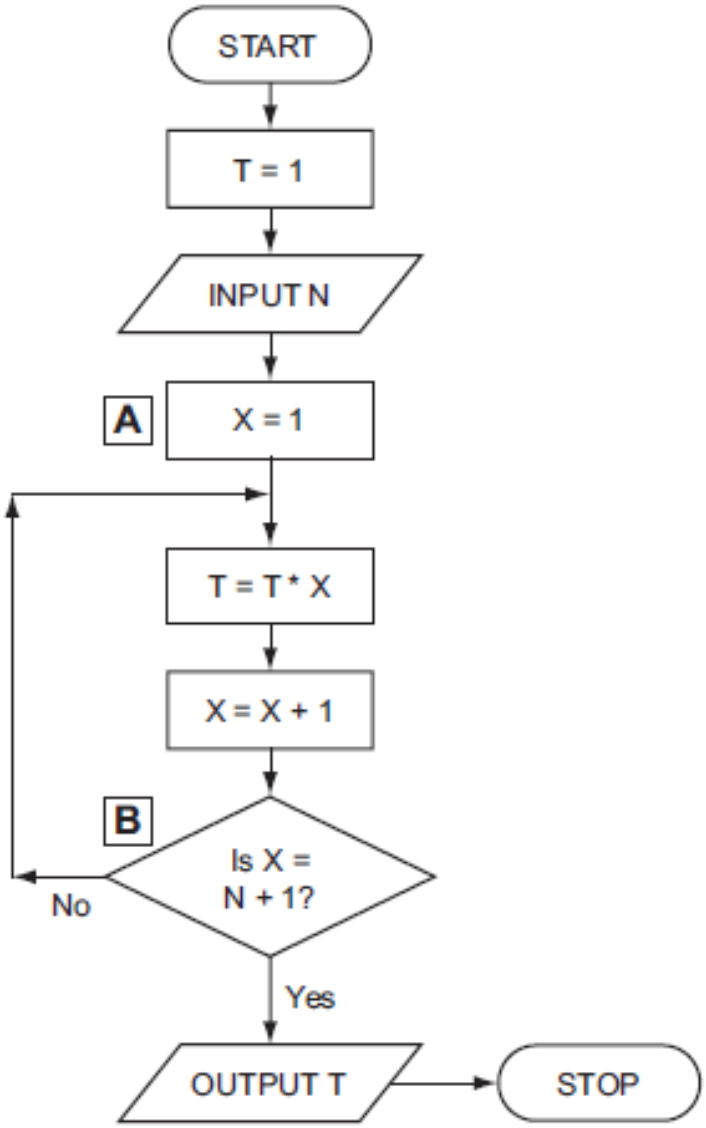
(a) Complete the following table for the three sets of input data. [3]

INPUT		OUTPUT
U	V	
5	5	
6	-6	
12	4	

(b) The above algorithm has been placed in a library of routines. Give one advantage of doing this.

……
……[1]

Q5: Summer 2009
Study the flowchart very carefully.



(a) Complete the table to show what outputs you would expect for the two inputs.[2]

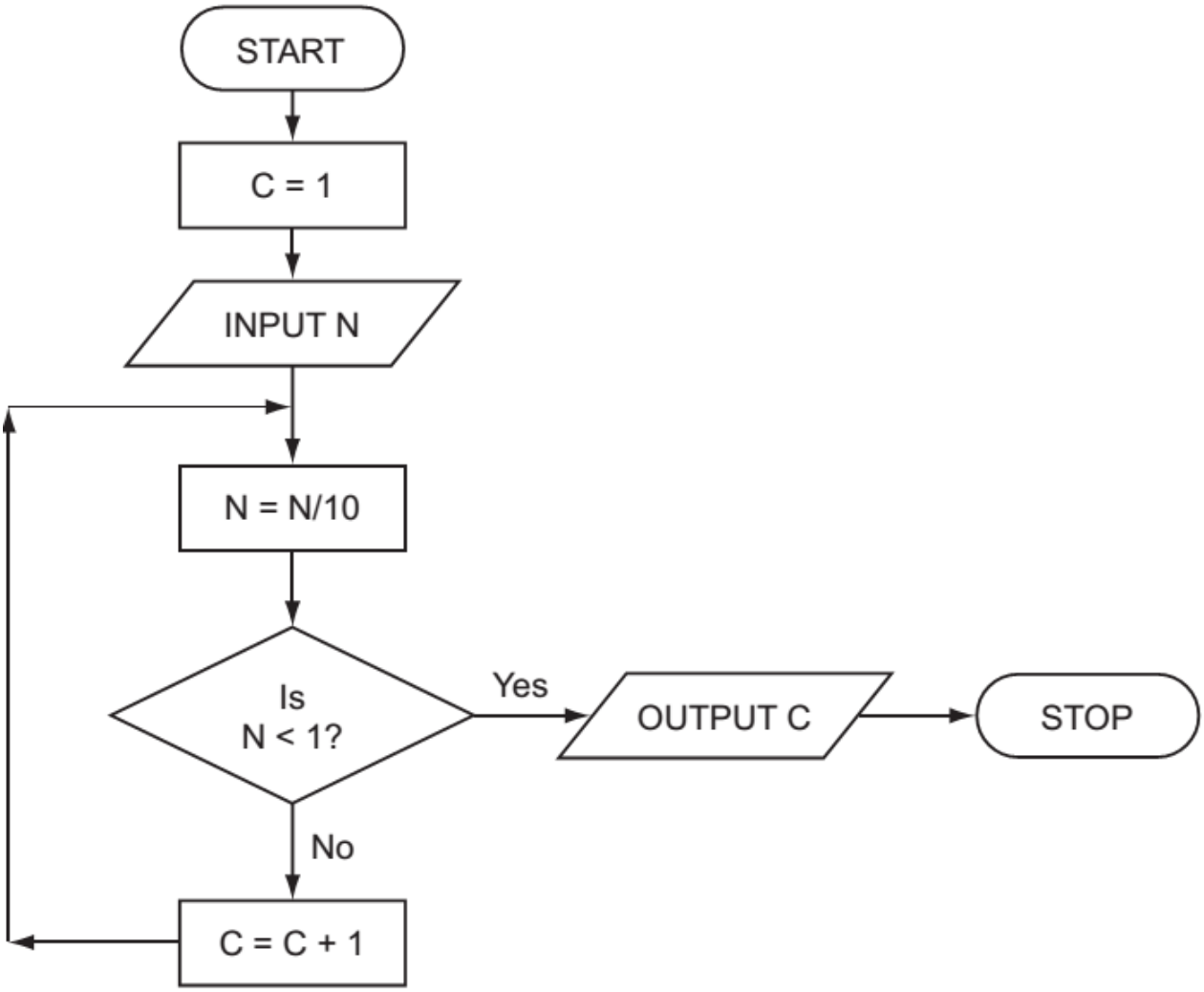
Input N	Output T
5	
1	

(b) Write down a possible LOOP construct for the section A to B in the flowchart using pseudocode.

……
……
…….. [2]

Q6: Winter 2009. P11

Study the flowchart.

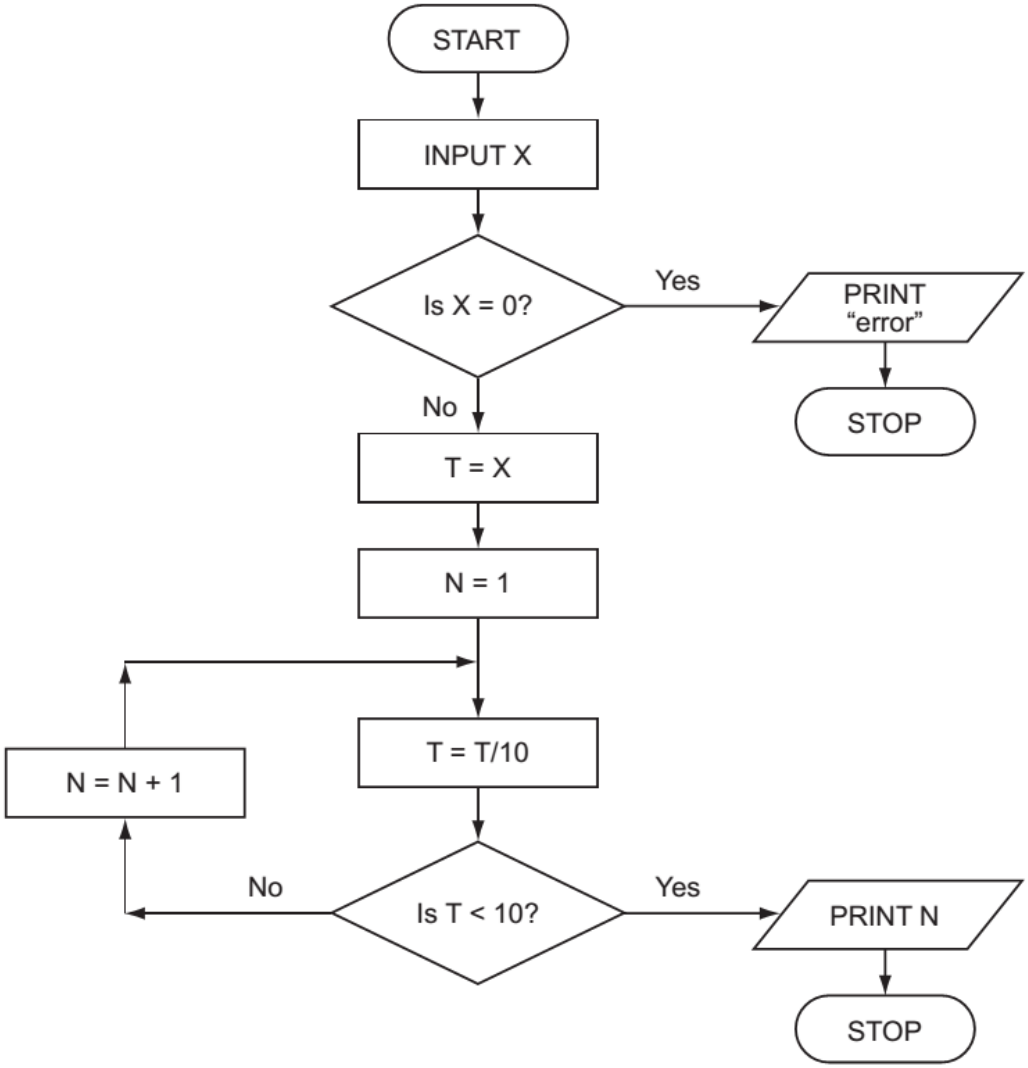


Complete the table to show what outputs you would expect for the three inputs. [3]

INPUT N	OUTPUT C
55	
2100	
1	

Q7: Summer 2010 P12

Study the following flowchart very carefully:

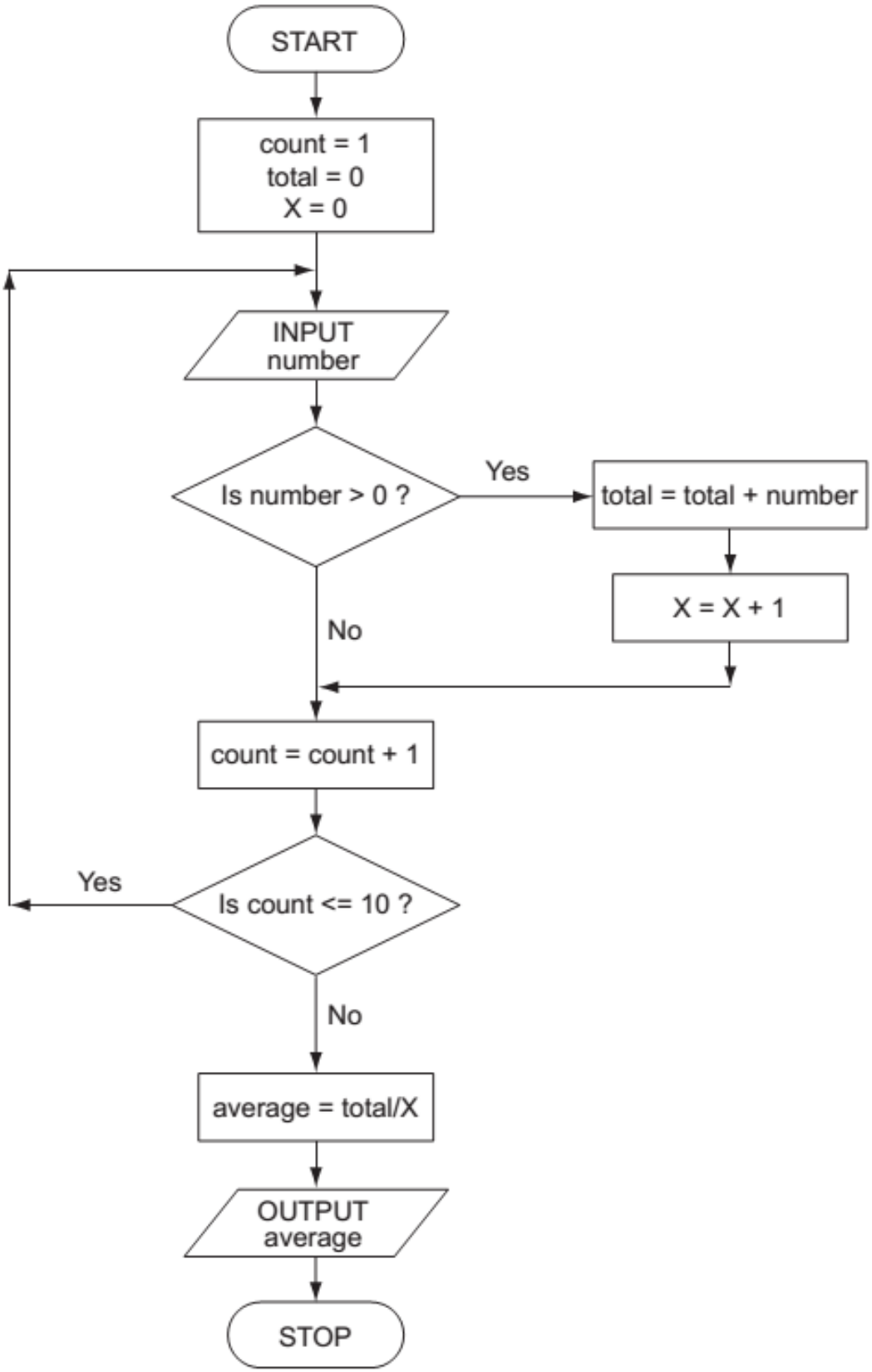


What output would you expect if the following data was input into the flowchart? [3]

X	OUTPUT
-150	
540	
0	

Q8:Summer 2011 P11

Study the following flowchart very carefully:



(a) Complete the trace table for the following data set:

15, -2, 0, 8, 0, 21, -8, -12, 1, 25 [4]

Count	Number	Total	X	Average	OUTPUT

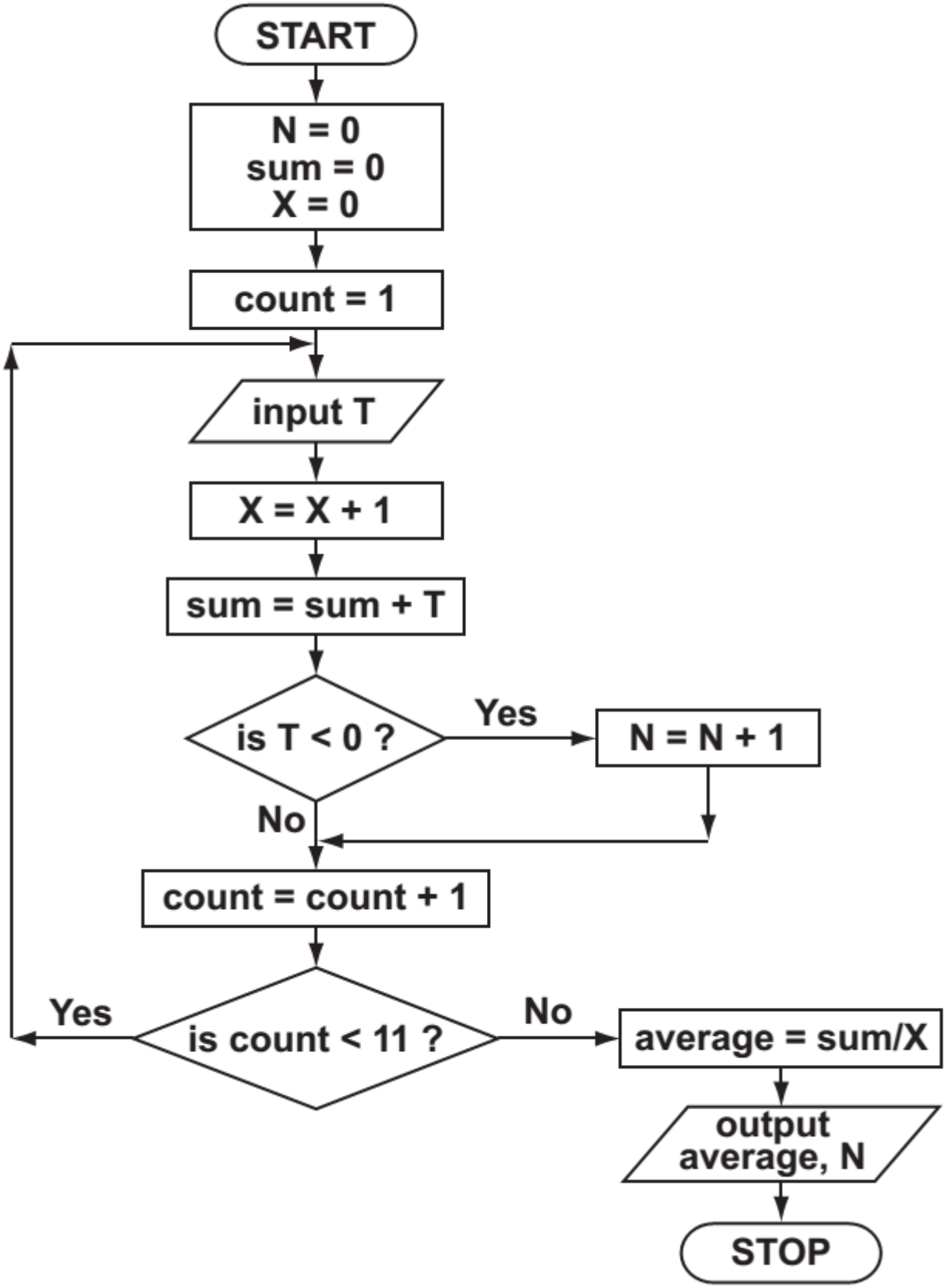
(b) What is the purpose of this flowchart?

.....

.....[1]

Q9: Summer 2011 P12

The following flowchart inputs ten temperatures andoutputs the average (mean) temperature and thenumber of temperatures which were negative (i.e. < 0).



(a) Complete the trace table for the following data set:

15, 11, 16, -4, -10, 8, 10, -3, 17, 10 [6]

N	Sum	X	Count	T	Average

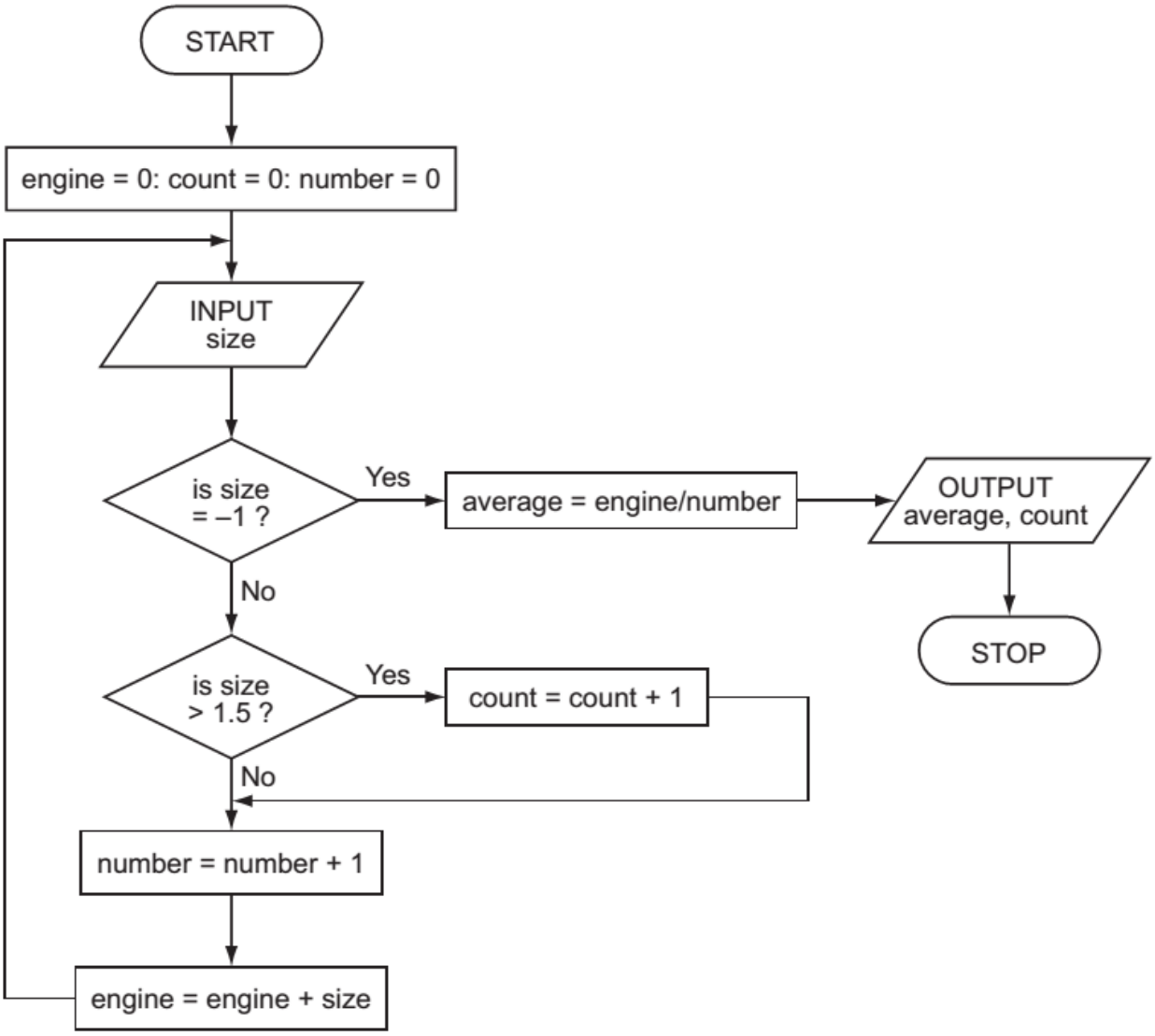
(b) What values is output from the flowchart using given test data?

_____ [1]

Q 10:Winter 2011 P13

The following flowchart inputs the size of a number of car engines; a value of -1 stops the input.

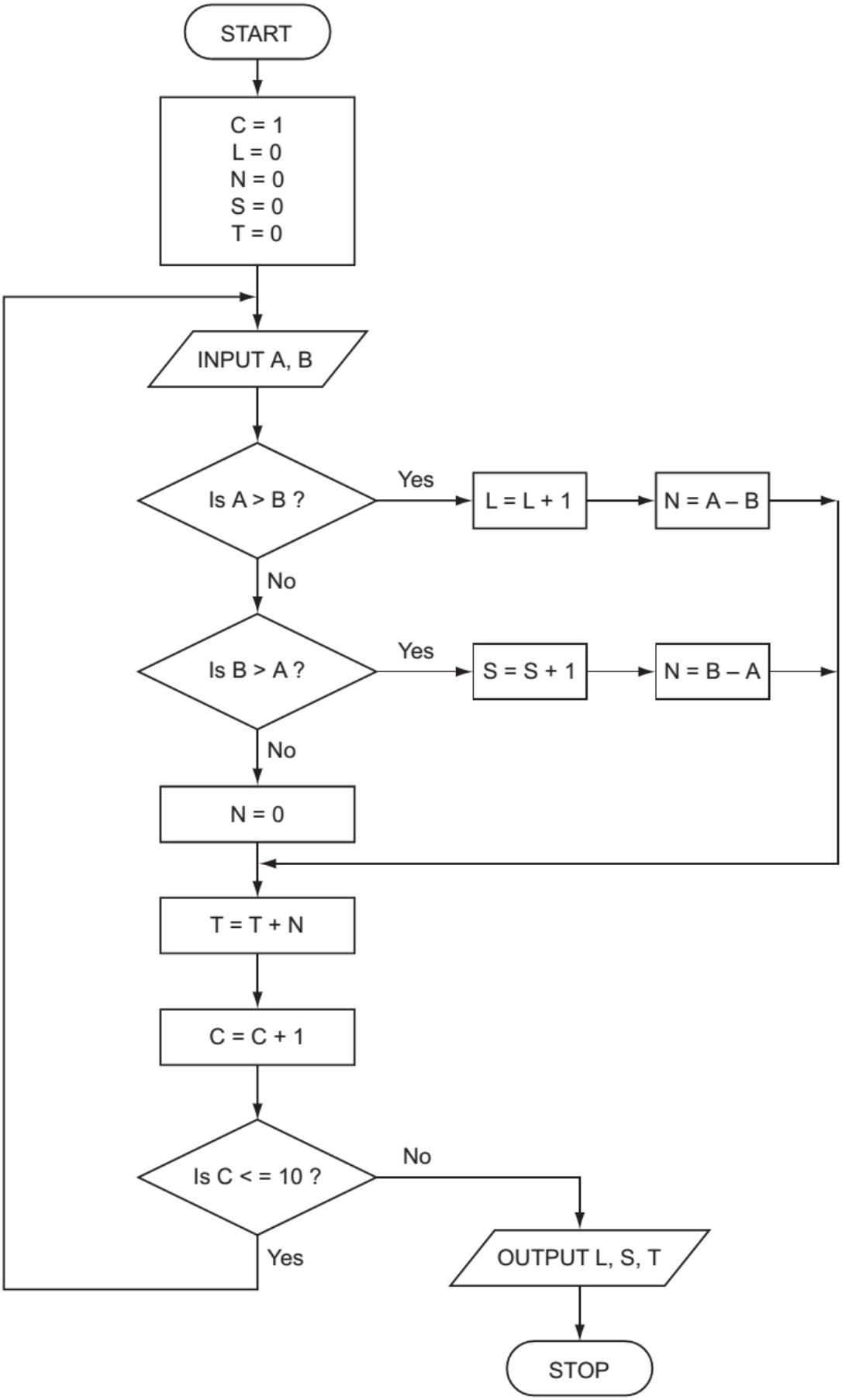
The following information is output: average engine size and number of engines with size>5.



Complete the trace table for the following input data:
1.8, 2.0, 1.0, 1.3, 1.0, 2.5, 2.0, 1.3, 1.8, 1.3, -1

engine	count	number	size	average	OUTPUT

Q 11: Summer 2012. P11



(a) Complete the trace table for the following data:
8, 4, 3, 1, 5, 8, 4, 2, 1, 3, 2, 2, 1, 2, 5, 5, 4, 0, 5, 4

C	L	N	S	T	A	B

[6]

(b) What is the final output from the algorithm?

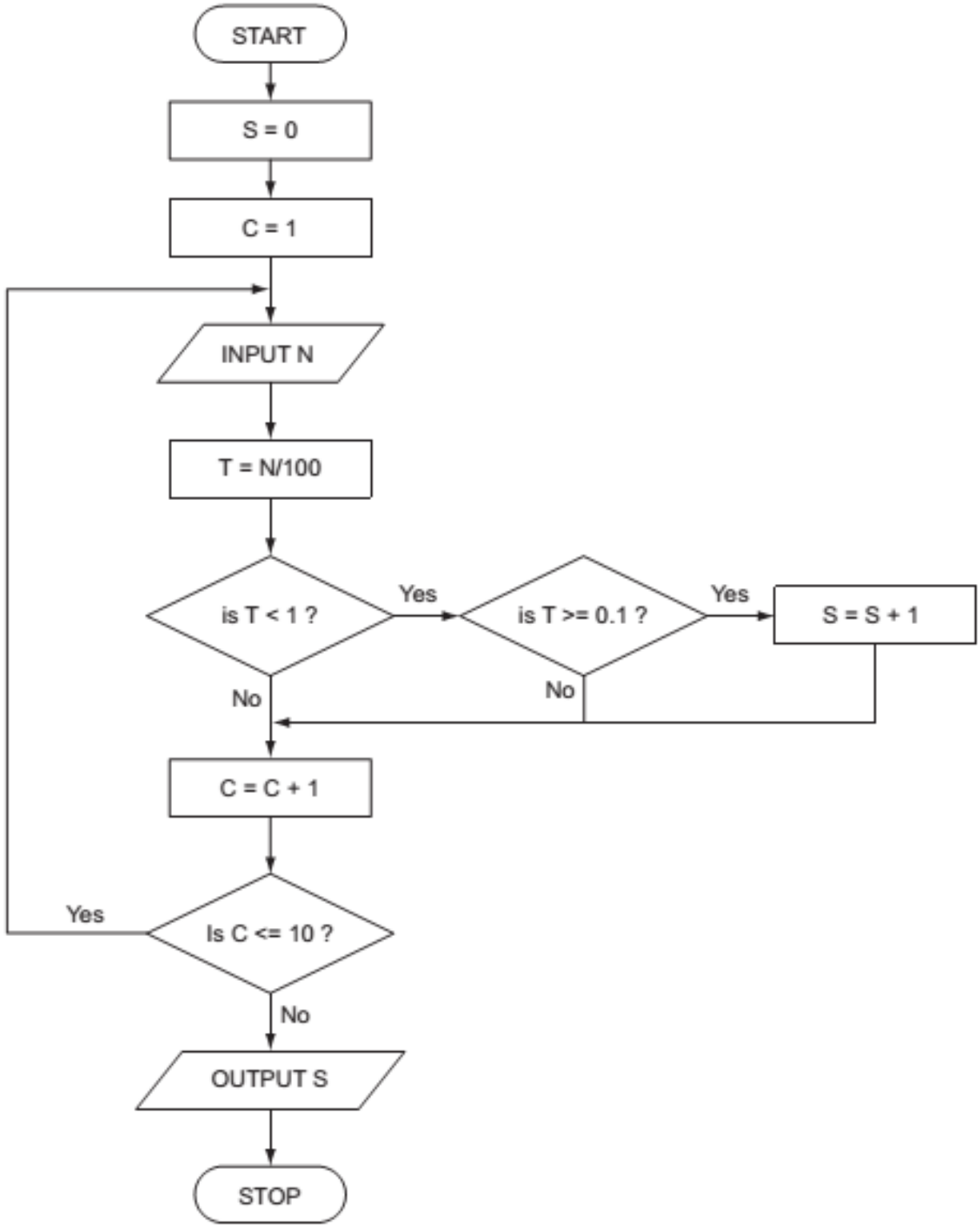
L = _____

S = _____

T = _____ [2]

Q12: Winter 2012 P12

Study the following flowchart very carefully.



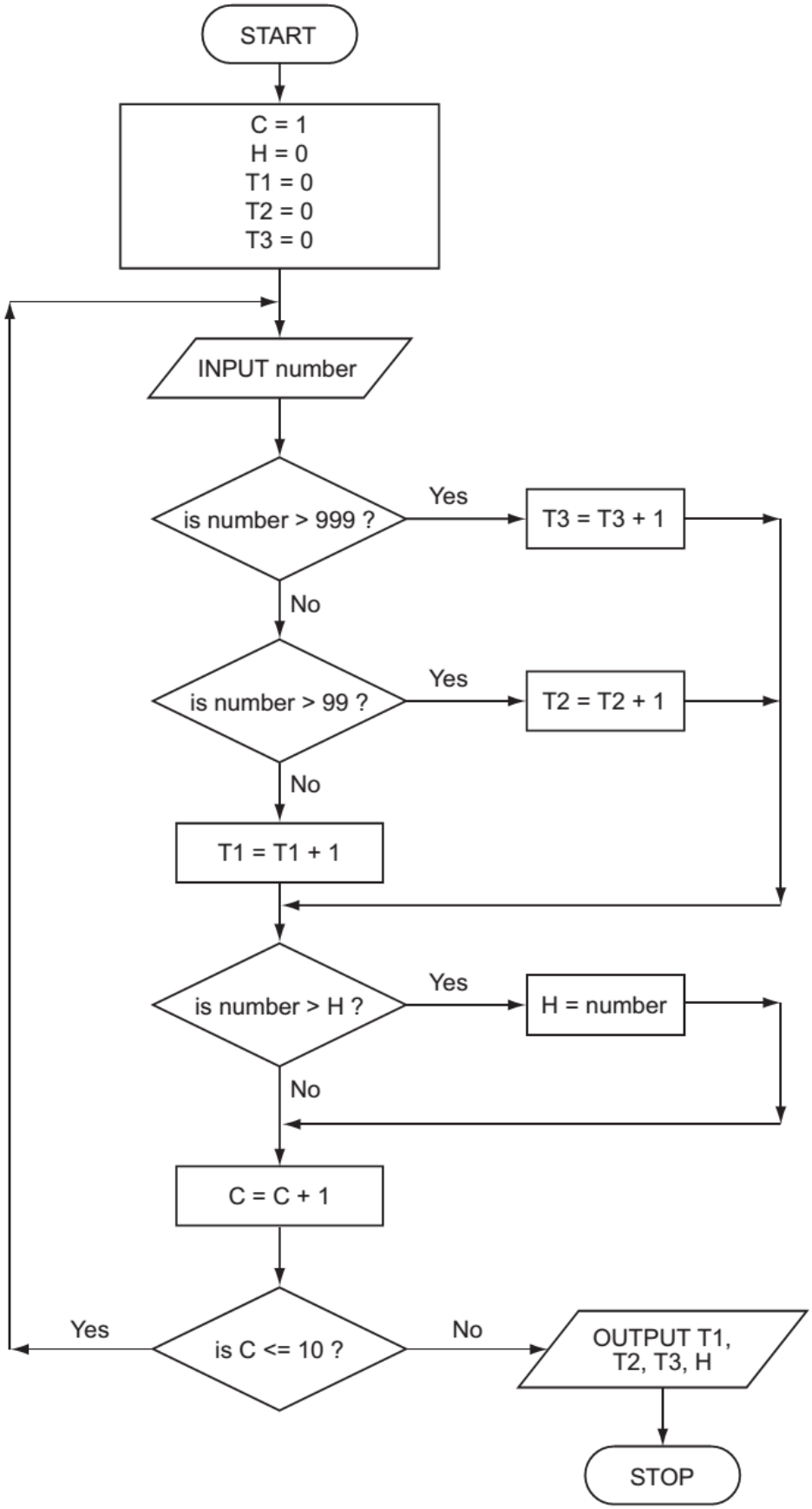
Complete the trace table for the following data:
15, 8, 251, 35, 60, 3, 2, 1516, 19, 55

S	C	N	T	OUTPUT

[5]

Q13: Winter 2012 P13

Study this flowchart very carefully.



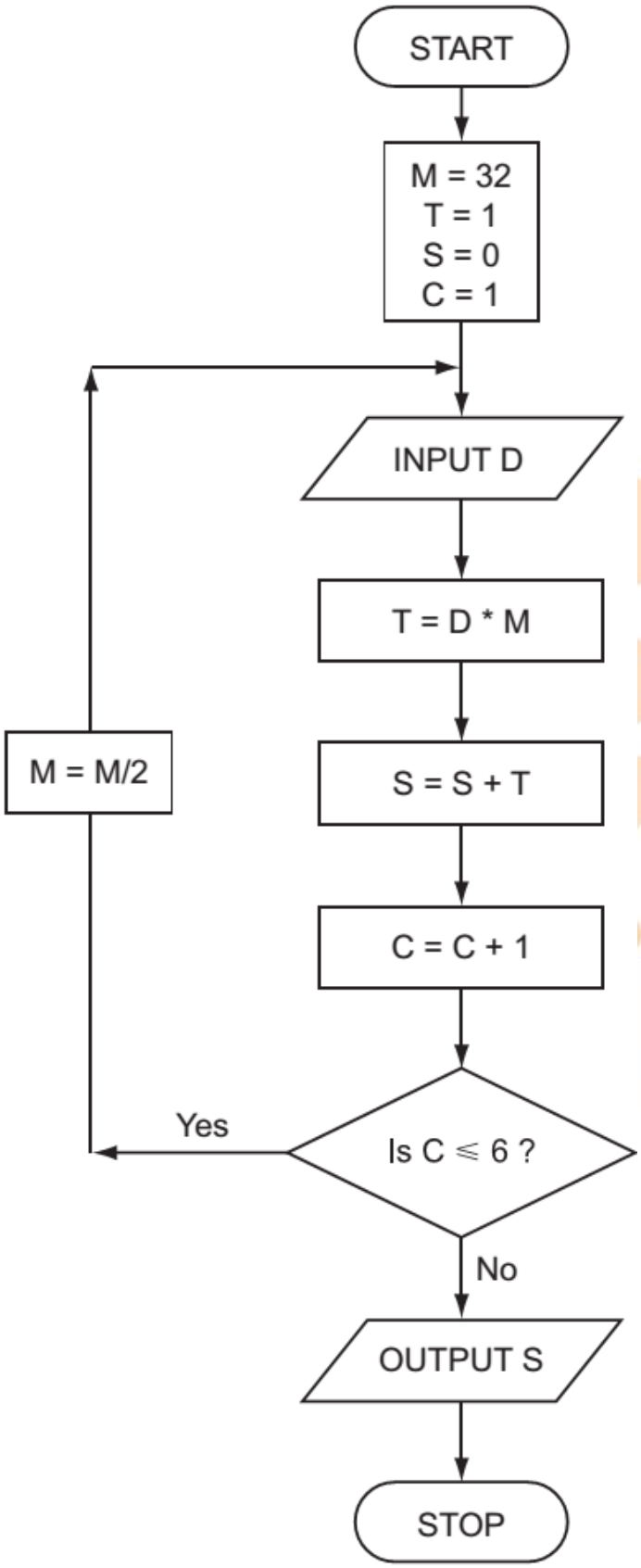
Complete the trace table for the following data:

1500, 1000, 100, 10, 999, 99, 2000, 5, -3, 0

C	H	T1	T2	T3	Number	OUTPUT

Q14:Summer 2012. P12

Carefully study the following flowchart:



(a) Complete the trace table for the following data: [4]

1, 0, 1, 1, 0, 1

M	T	S	C	D

(b) What process does this flowchart perform?

……

……

…… [1]

(c) Predict the output from the flowchart for an input of 1, 1, 1, 1, 0, 0

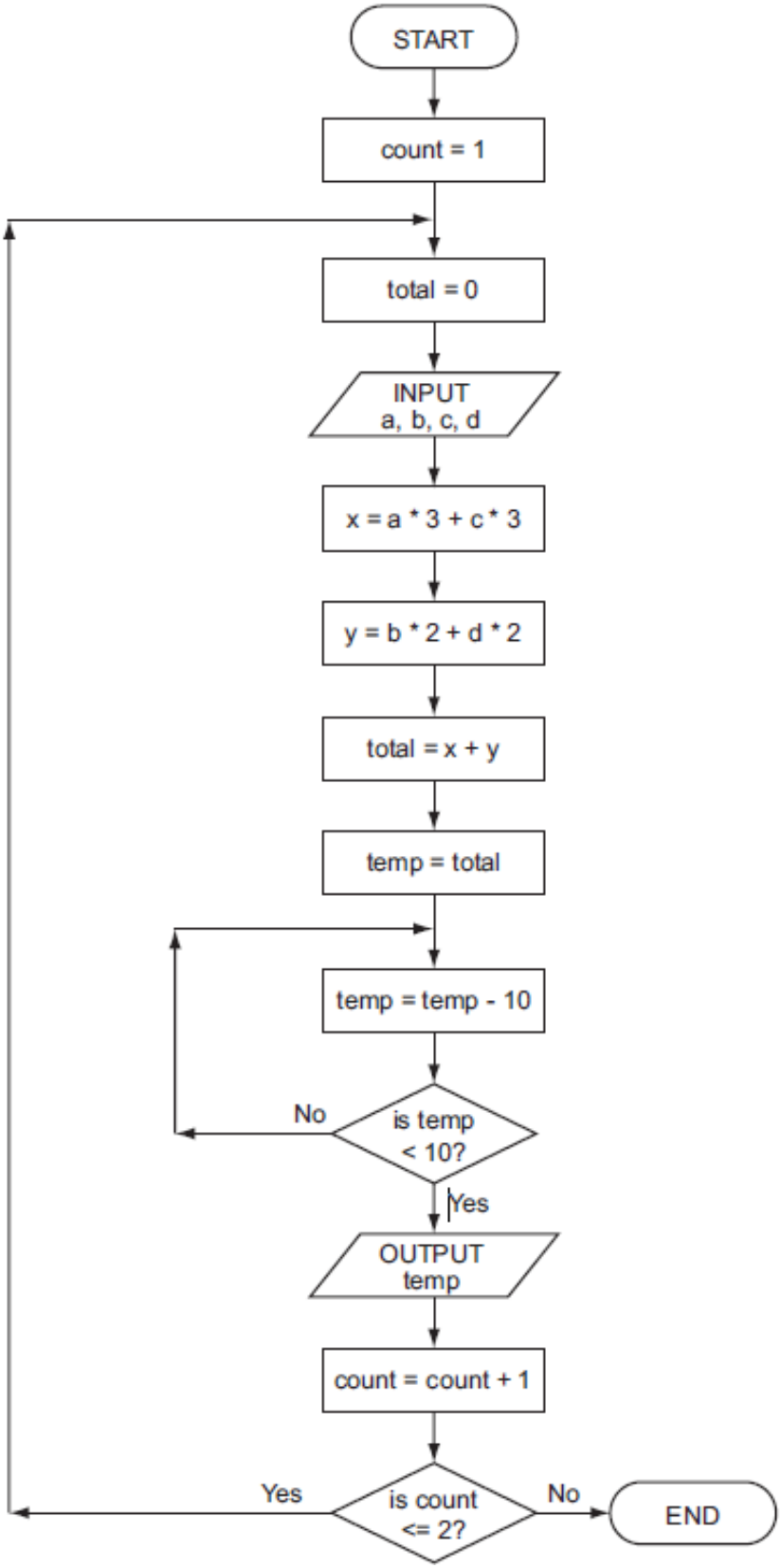
……

……

…… [1]

Q15: Winter 2013. P12

Study the following flowchart very carefully

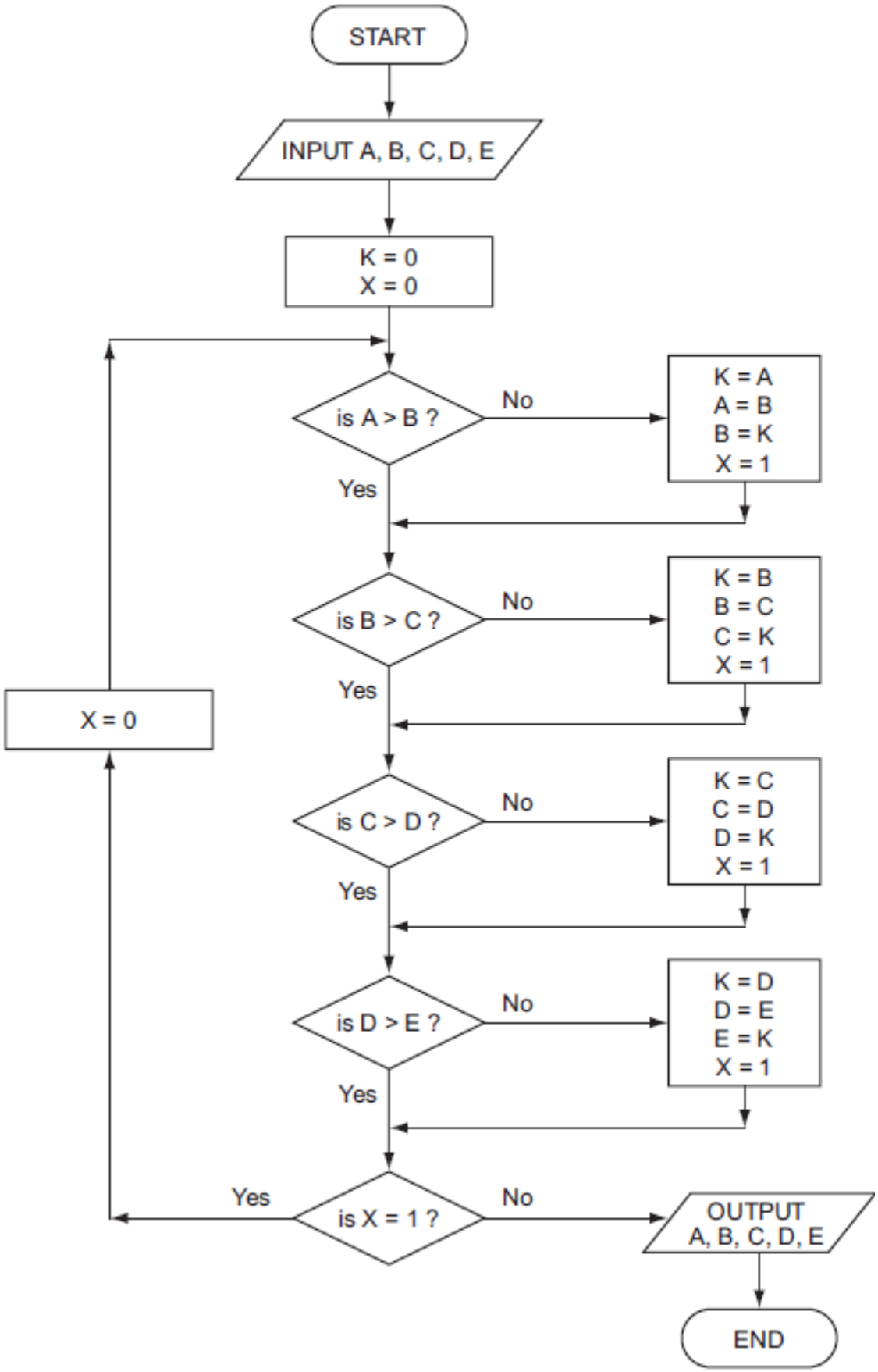


Complete the trace table for the following two sets of data:
(i) a = 5, b = 4, c = 1, d = 9
(ii) a = 5, b = 9, c = 4, d = 1

count	Total	a	b	c	d	X	y	temp	OUTPUT

Q 16:Summer 2013. P12

Study the following flowchart very carefully:



(a) Complete the following trace for this flowchart using the following test data:

3, 5, 1, 4, 8

A	B	C	D	E	K	X

b) What values are outputs from the following flowchart using the above test data?

.....

(c) What function is this flowchart carrying out?

.....

.....

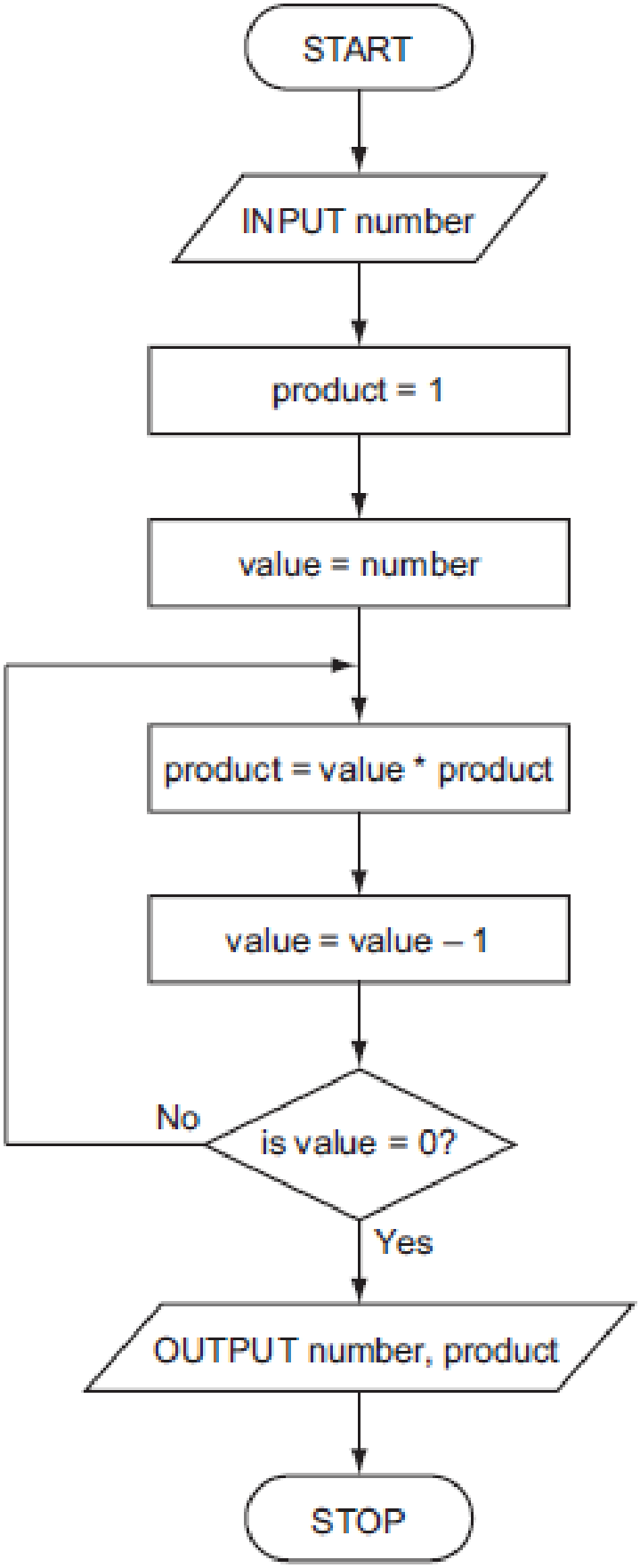
(d) What would happen if the value X wasn't set to 0 in the return loop of the flowchart?

.....

.....

Q17: Summer 2014. P11

Study the following flowchart very carefully.

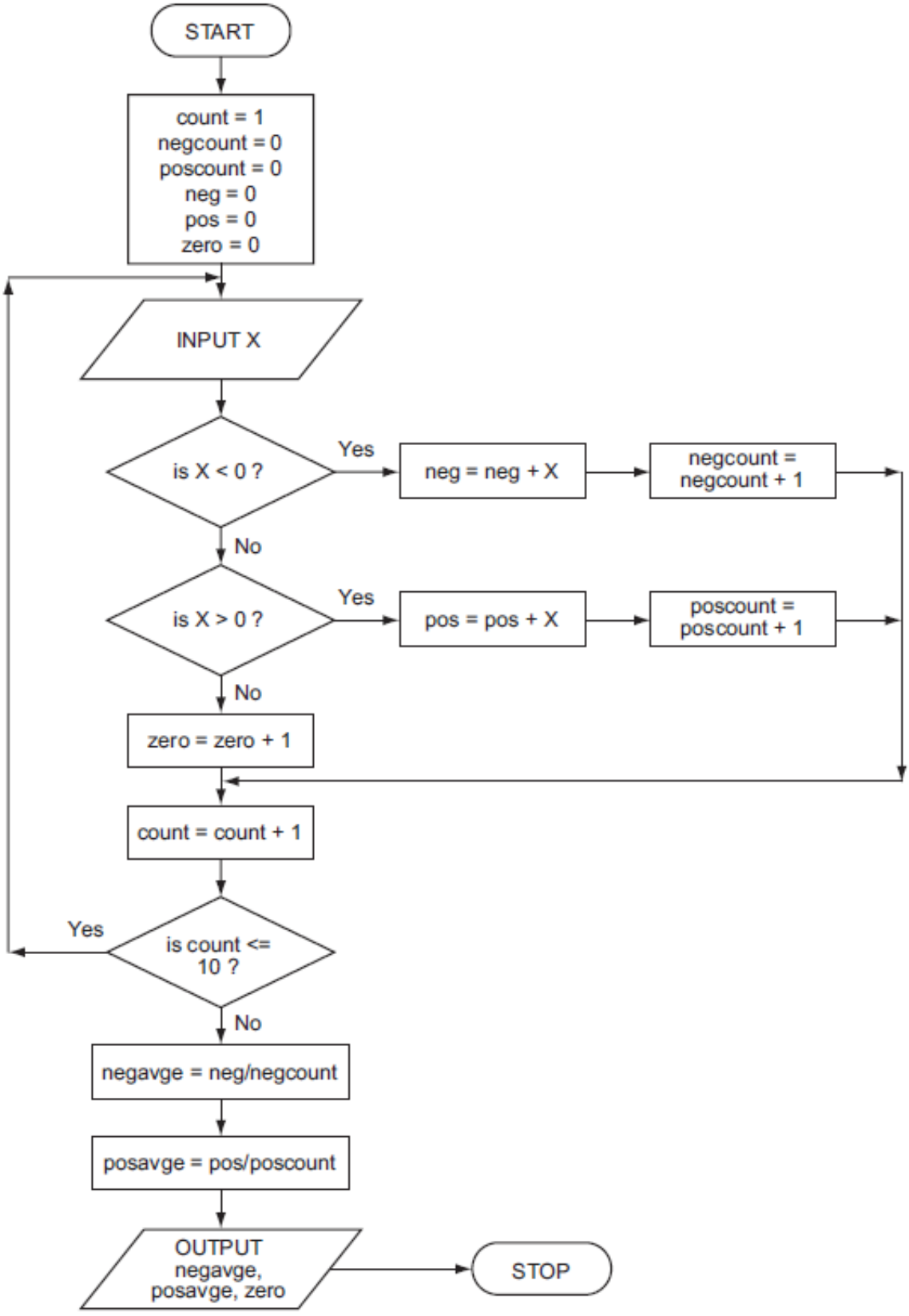


Complete the trace table for the input value of 5:

Number	Product	Value	OUTPUT

Q 18: Summer 2014. P12

Study the following flowchart very carefully.

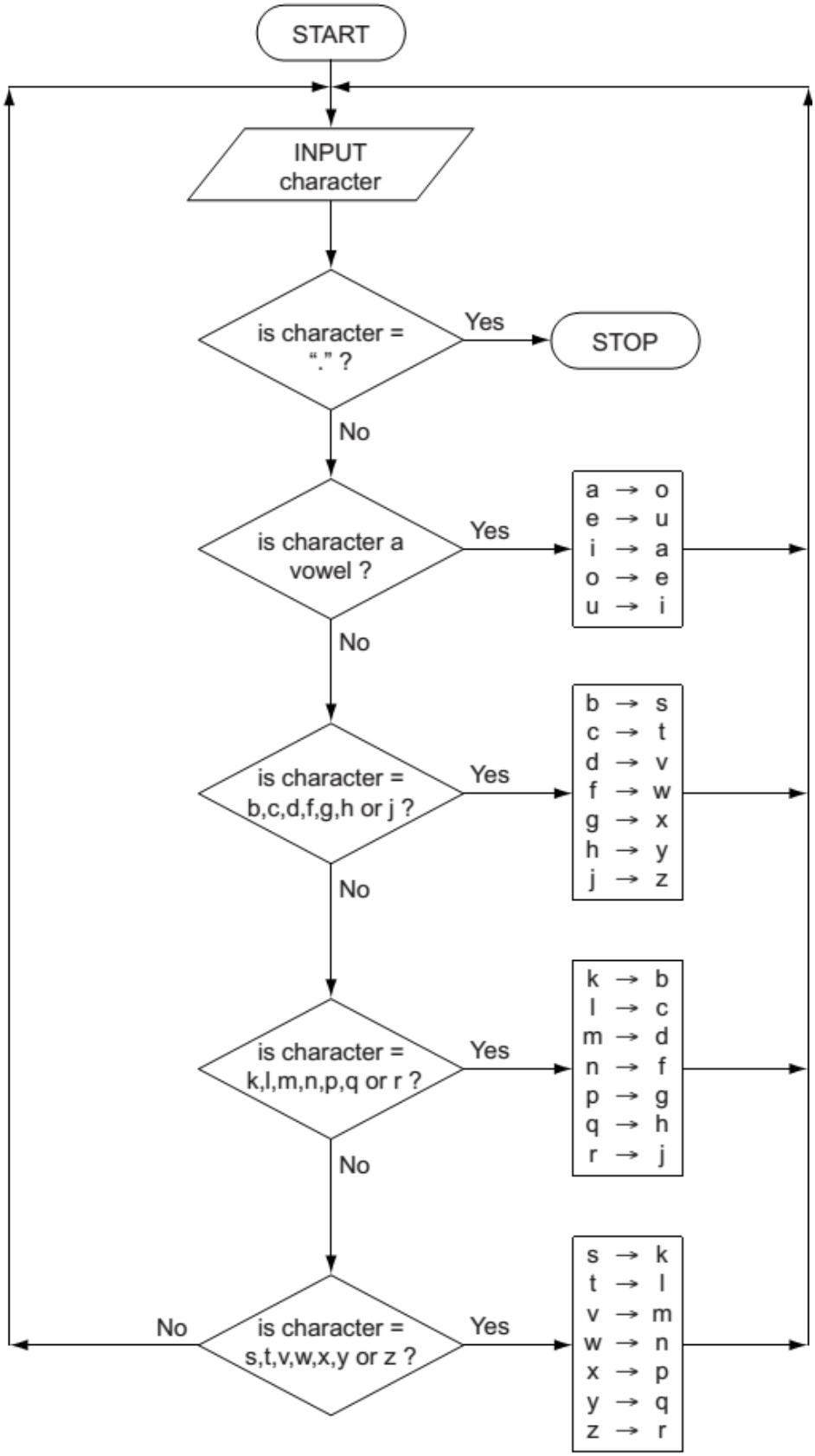


Complete the trace table for the flowchart using the following data:
0, 3, 5, 6, -4, -1, 0, 0, -4, 10

negcount	poscount	neg	pos	zero	count	X	negavge	posavge

Q 19: Winter 2014 P13

Data sent across the Internet are frequently encrypted. The following flowchart shows a basic encryption method (Note: the □ symbol in the flowchart means "is replaced by").



For example,

h	e	l	l	o		i		a	m		a		r	o	b	o	t	.

becomes:

y	u	c	c	e		a		o	d		o		j	e	s	e	l	.

(a) flowchart to encrypt the following message: Use the

m	e	e	t	i	n	g		w	i	l	l		g	o		a	h	e	a	d	.

[2]

(b) Use the information in the flowchart to show which input message produced the following encrypted message:

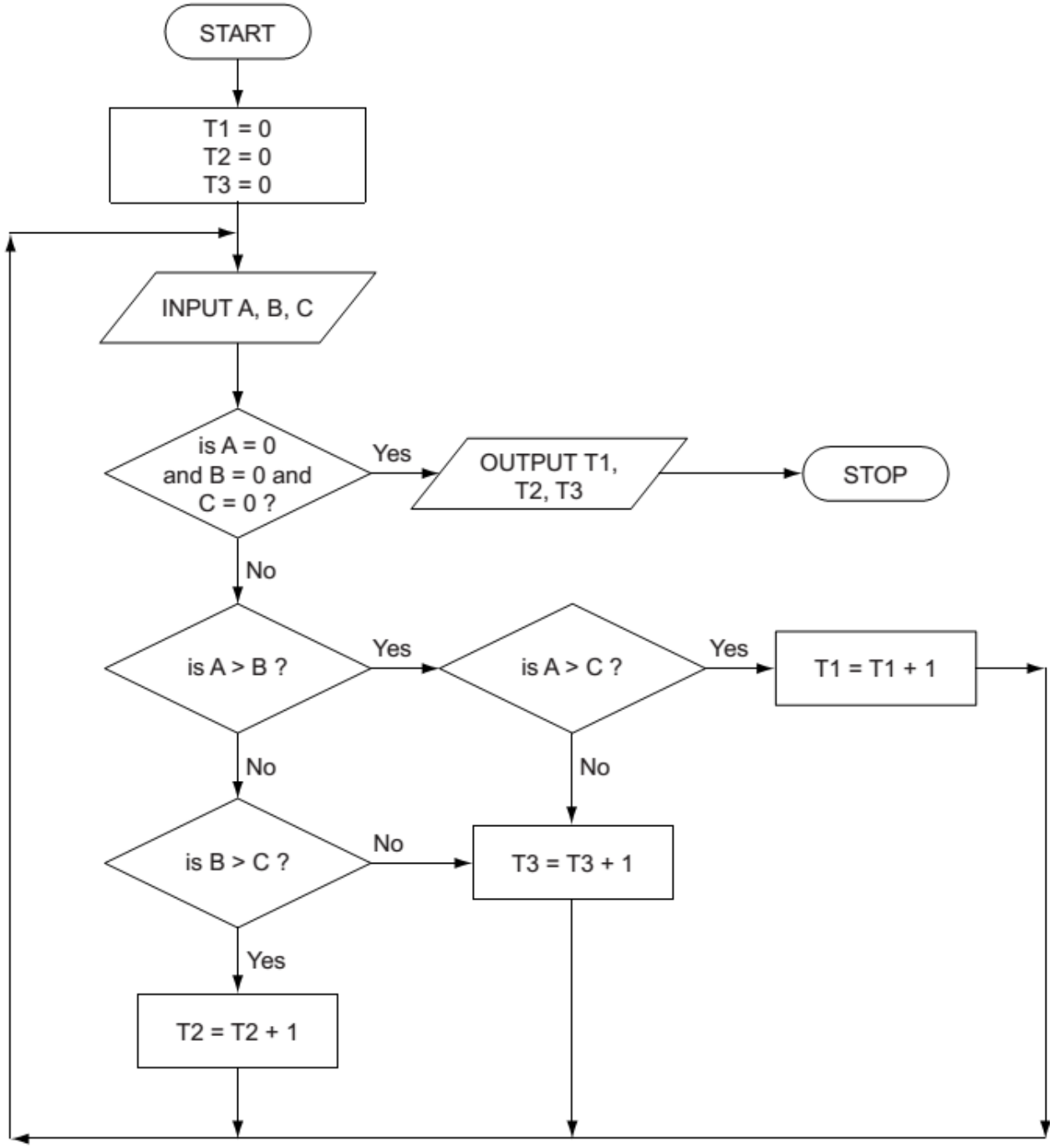
t	e	d	g	i	l	a	f	x		a	k		w	i	f	.

[2]

(c) Many customers shop online.
Apart from encryption, describe **three** other security features built into many online shopping websites.

1 .....
.....
2 .....
.....
3 .....
.....[3]

Q20: Winter 2014 P13



(a) Complete the trace table for the flowchart using the following data:

3, 2, 1 4, 8, 7 6, 0, 3 5, 6, 9 4, 11, 3 0, 0, 0

T1	T2	T3	A	B	C	OUTPUT

(b) This flowchart does not give correct answers for certain sets of test data.
Suggest a data set that would give an incorrect answer.
Give a reason for your choice.

data set

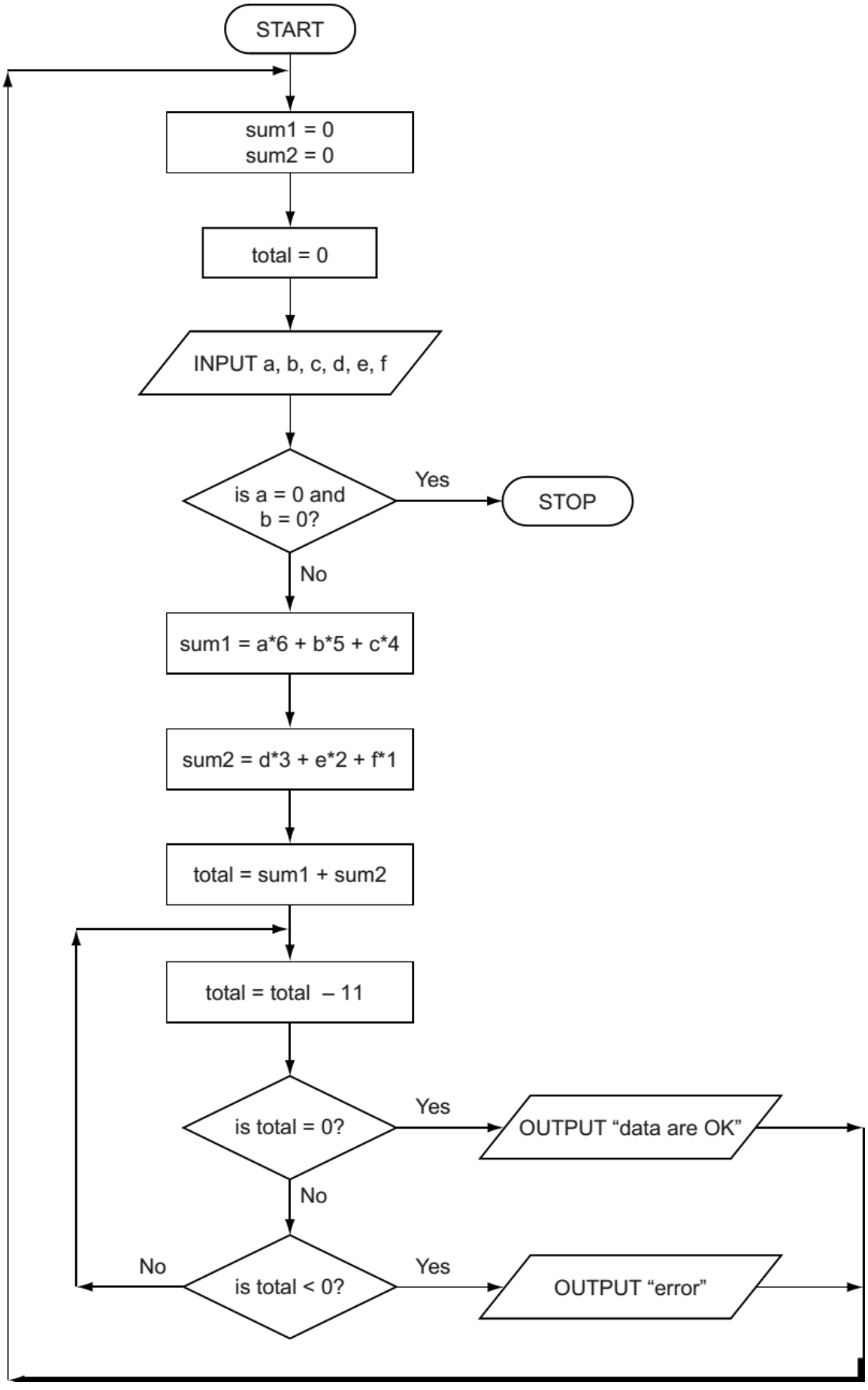
.....
.....

reason

.....
..... [2]

Q21: Winter 2014 P12

Study the following flowchart very carefully.



Complete the trace table for this flowchart using the following data:

4, 3, 2, 0, 0, 8

5, 0, 1, 2, 3, 4

0, 0, 0, 0, 0, 0

sum1	sum2	total	a	b	c	d	e	f	OUTPUT

Marking Schemes

Q1) (a) 1 [1](b) 10,5,16,8,4,2,1 one mark one mark [2]
Q2)(a) 2, 4, 1 [3]
(b) (i) Any one point from:
computer check on input data check data is wrong/correct = 0
detects any data which is incomplete or not reasonable
(ii) Any one point from:
length check – e.g. only 30 characters in name field
character check – e.g. name doesn't contain numeric chars
range check – e.g. day of month in date is between 1 and 31
format check – e.g. date in the form xx/yy/zz
check digit – e.g. end digit on bar code to check if it is valid
type check – e.g. integer, real
(presence check = 0) [2]
Q3) (a) H M
18 15
18 40 [2]
(b) Any one point from:
M would become 60 and should be 0 for correct time
H would become 18 and should be 19 for correct time [1]
(c) Would get a negative answer for H [1]
Q4)9 (a) 2.5, Error, 3 [3
(b) Any one from:
would be fully tested
doesn't need to be re-written each time section of program needed [1]

Q5) (a) 120
1 [2]
(b) FOR X = 1 TO N + 1
(T = T * X)
NEXT X
OR
REPEAT
(T = T * X)
X = X + 1
UNTIL X = N + 1
OR
WHILE X<> N + 1 DO
(T = T * X)
X = X + 1
ENDWHILE
(1 mark for correct first line of loop construct)
(1 mark for correct loop control and last line of loop construct) [2]

Q6) 2
4
1 [3]

Q7) Expected output:
1
2
Error

Q8)

Count	Number	Total	X	Average	OUTPUT
1		0	0		
2	15	15	1		
3	-2				
4	0				
5	8	23	2		
6	0				
7	21	44	3		
8	-8				
9	-12				
10	1	45	4		
11	25	70	5	14	14

(b) Find the average of all positive numbers entered [1]

Q9) (a)

N	Sum	X	Count	T	Average
0	0	0	1		
	5	1	2	5	
	16	2	3	11	
	32	3	4	16	
1	28	4	5	-4	
2	18	5	6	-10	
	26	6	7	8	
	36	7	8	10	
3	33	8	9	-3	
	50	9	10	17	
	60	10	11	10	
					6

(b) 6,3

Q10)

Engine	Count	Number	Size	Average	Output
0	0	0	1.8		
1.8	1	1	2.0		
3.8	2	2	1.0		
4.8		3	1.3		
6.1		4	1.0		
7.1	3	5	2.5		
9.6		6	2.0		
11.6	4	7	1.3		
12.9	5	8	1.8		
14.7		9	1.3		
16.0		10	(-1)		
				1.6	
					1.6, 5

Q11)

C	L	N	S	T	A	B
1	0	0	0	0	8	4
2	1	4		4	3	1
3	2	2		6	5	8
4		3	1	9	4	2
5	3	2		11	1	3
6		2	2	13	2	2
7		0		13	1	2
8		1	3	14	5	5
9		0		14	4	0
10	4	4		18	5	4
11	5	1		19		

(b) L=5, S=3, T=19

Q13)

C	H	T1	T2	T3	Number	Output
1	0	0	0	0	1500	
2	1500			1	1000	
3				2	100	
4			1		10	
5		1			999	
6			2		99	
7		2			2000	
8	2000			3	5	
9		3			-3	
10		4			0	
11		5				
						5,2,3,2000

Q14)

M	T	S	C	D
32	1	0	1	1
16	32	32	2	
8	0	32	3	0
4	8	40	4	1
2	4	44	5	1
1	0	44	5	1
	1	45	7	1

(b) converting binary number into equivalent base 10 number [1]
(c) 60 [1]

Q16)

A	*B*	*C*	*D*	*E*	*K*	*X*
3	*5*	*1*	*4*	*8*	*0*	*0*
5	*3*				*3*	*1*
		4	*1*		*1*	*1*
			8	*1*	*1*	*1*
						0
	4	*3*			*3*	*1*
		8	*3*		*3*	*1*
						0
	8	*4*			*4*	*1*
						0
8	*5*				*5*	*1*
						0

(b) 8, 5, 4, 3, 1 [1]
(c) SORT/ORDER (descending) routine [1]
(d) – would continue looping round even when sorting complete
– loop would never end/infinite loop [1]

Q17)

Number	*Product*	*Value*	*OUTPUT*
5	*1*	*5*	
(5)	*5*	*4*	
(5)	*20*	*3*	
(5)	*60*	*2*	
(5)	*120*	*1*	
(5)	*(120)*	*0*	
			5, 120

Q18) (NOTES: Additional 0s in any column (UNLESS THEY ARE JUST THE REPEAT OF 0 VALUES) lose the mark for that column.
If columns 1 to 7 are wrong there can be one mark for initialisation (0 0 0 0 0 1) and a mark forthe correct output -3, 6).

negcount	**poscount**	**neg**	**pos**	**zero**	**count**	**X**	**negavge**	**posavge**
0	0	0	0	0	1			
				1	2	0		
	1		3		3	3		
	2		8	4	5			
	3		14		5	6		
1		-4			6	-4		
2		-5			7	-1		
				2	8	0		
				3	9	0		
3		-9			10	-4		
	4		24		11	10		
							-3	6

Q19)

(a)

d	u	u	l	a	f	x		n	a	c	c		x	e		o	y	u	o	v	.

<-------- 1 mark --------> <-------- 1 mark --------> [2]

(b)

c	o	m	p	u	t	i	n	g		i	s		f	u	n	.

<-------- 1 mark --------> <-------- 1 mark --------> [2]

(c) Any three from:

– customers need a password / PIN
– use of card readers / use of Transaction Authentication Number (TAN)
– only certain characters from password / PIN requested…
– …the requested characters change each time user logs on
– card security code requested
– use of drop down boxes
– use of a customer reference number
– inform customer when they last logged on to the website
– use of image verification code e.g. CAPTCHA
– make reference to something unique to the customer e.g. their mobile phone number
– use of secure protocol e.g. https, padlock symbol[3]

Q20)

T1	T2	T3	A	B	C	OUTPUT
0	0	0				
			3	2	1	
1						
	1		4	8	7	
2			6	0	3	
		1	5	6	9	
	2		4	11	3	
			0	0	0	
						2, 2, 1

If no marks are awarded for the columns then 1 mark can be given for correct initialisation ofT1, T2 & T3 as shown in the first row above. [5]
(b) – any data set (except 0, 0, 0) where 2/3 of the numbers are the same e.g. 2, 8, 8
– flowchart does not allow for numbers which have the same value [2]

Q21) NOTE: sum1, sum2 and total MUST be initialised for all three inputs to get the mark; allow repetition in any of the columns EXCEPT the OUTPUT column (e.g. sum1 can be 0, 47, 47, 47, 47, 47, 47);

sum1	sum2	total	a	b	c	d	e	f	OUTPUT
0	0	0	4	3	2	0	0	8	
47	8	55							
		44							
		33							
		22							
		11							
		0							Data are ok
0	0	0	5	0	1	2	3	4	
34	16	50							
		39							
		28							
		17							
		6							
		-5							Error
0	0	0	0	0	0	0	0	0	

Fill in missing statements

Questions are most commonly set from steps of bar code reading, sensors, operating ATM machine etc.

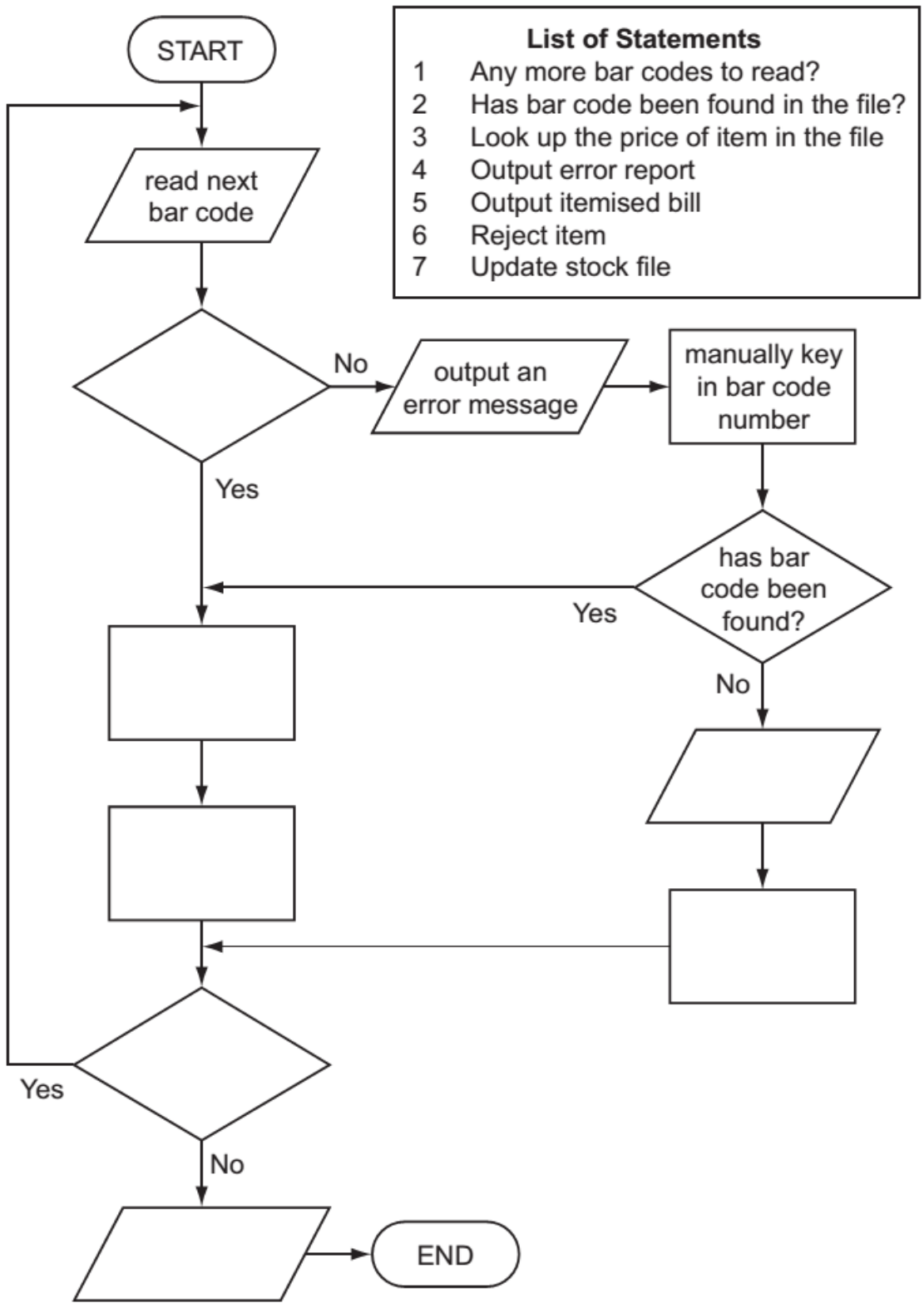
Steps of bar code reading:

When barcode has been read, then what happens?

- The barcode number is looked up in the stock database (the barcode is known as the **KEY FIELD** in the stock item record); this key field uniquely identifies each stock item.
- When the barcode number is found, the stock item record is looked up.
- The price and other stock item details are sent back to the checkout (or **POINT OFSALE TERMINAL (POS)**).
- The number of stock items in the record is reduced by one each time the barcode is read.
- This new value for number of stock items is written back to the stock item record.
- The number of stock items is compared to the re-order level; if it is less than or equal to this value, more stock items are *automatically* ordered.
- Once an order for more stock items is generated, a flag is added to the record to stop re-ordering every time the stock item barcode is read.
- When new stock items arrive, the stock levels are updated in the database.

Q1: Winter 2006

The following flowchart shows how the bar code written on an item is used to find the price, do stock control and produce an itemised bill. Select statements from the list below to complete the flowchart.



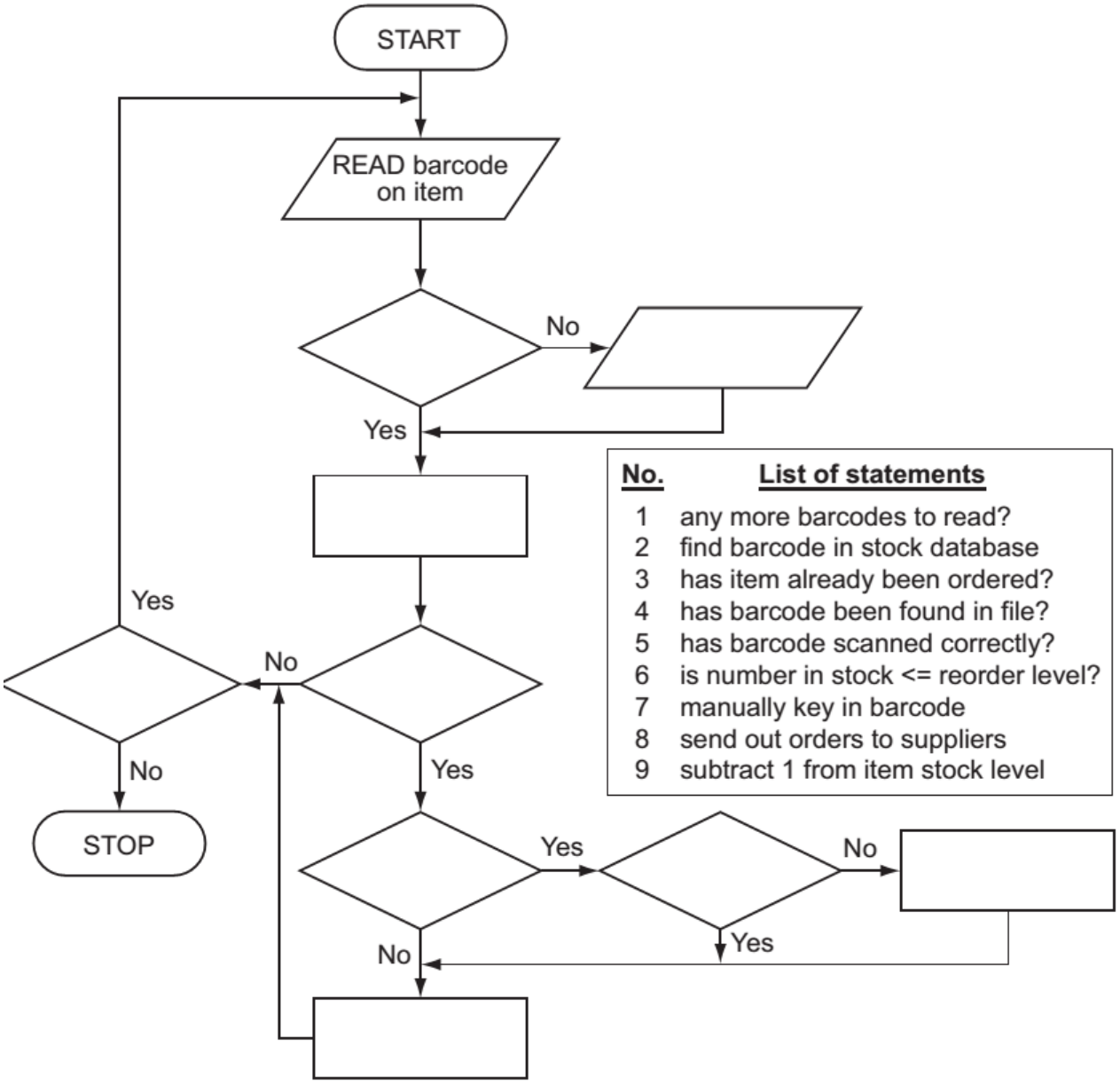
List of Statements

1. Any more bar codes to read?
2. Has bar code been found in the file?
3. Look up the price of item in the file
4. Output error report
5. Output itemised bill
6. Reject item
7. Update stock file

Q2: Winter2011 P11

The following flowchart shows how barcodes are used at the point of sale in an automatic stock control system.

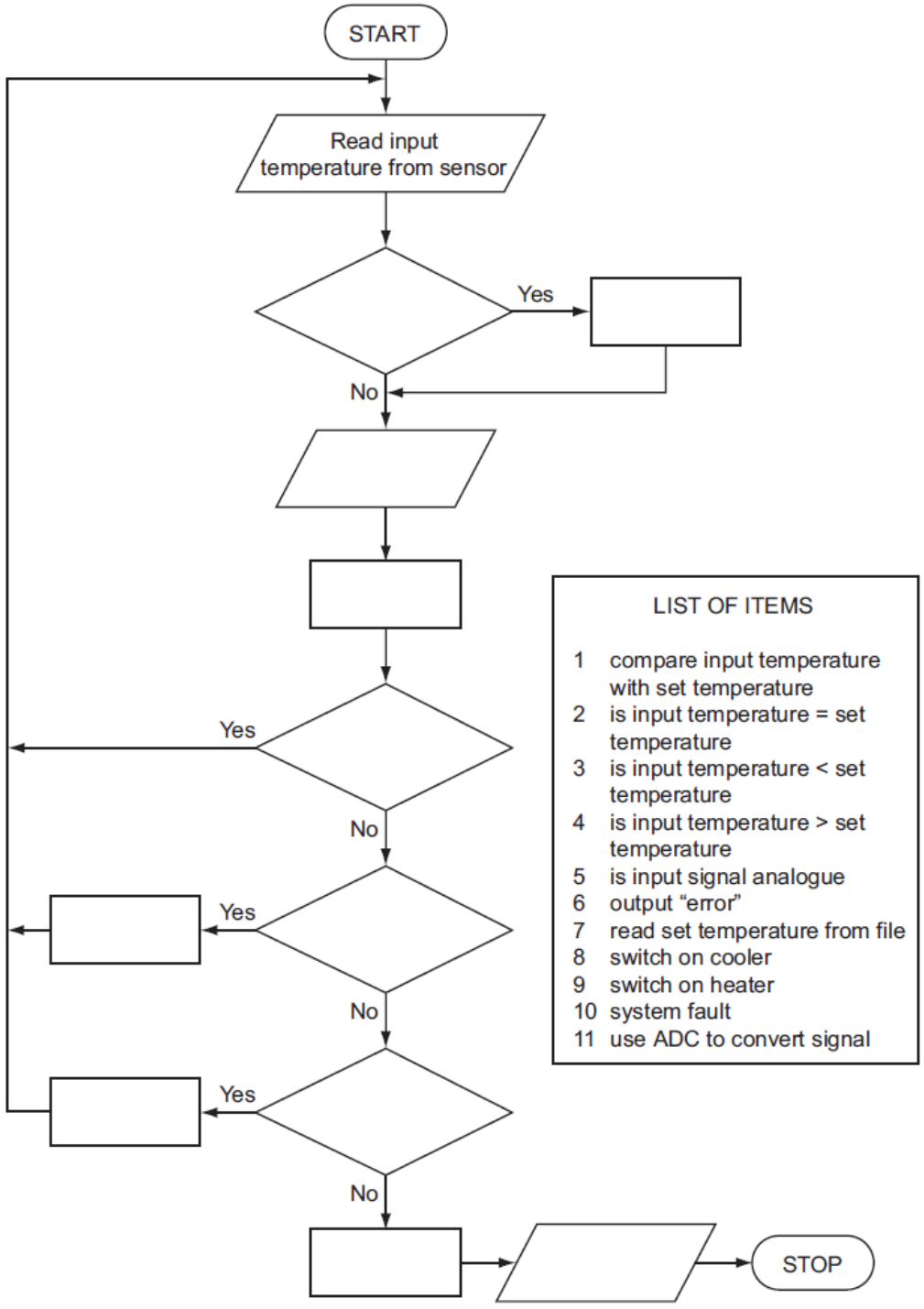
Select statements from the list below, using numbers only, to complete the flowchart.



No.	List of statements
1	any more barcodes to read?
2	find barcode in stock database
3	has item already been ordered?
4	has barcode been found in file?
5	has barcode scanned correctly?
6	is number in stock <= reorder level?
7	manually key in barcode
8	send out orders to suppliers
9	subtract 1 from item stock level

Q3: Winter 2008

The following flowchart shows how sensors (which can be analogue or digital) and a computer are used to control the temperature of a greenhouse for plants. Complete the flowchart using the items from the list below.

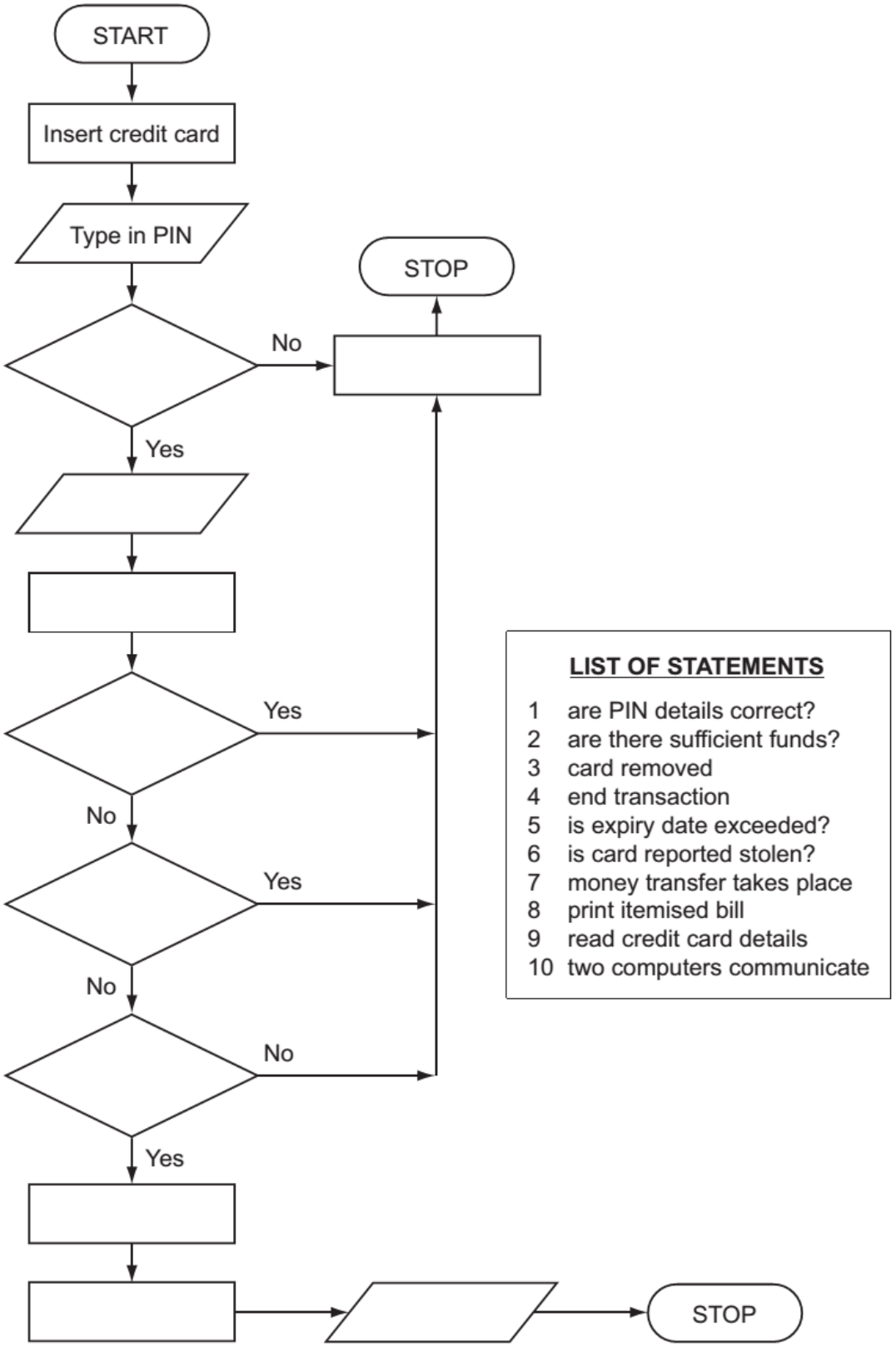


[6]

Q 4Summer 2009

The following flowchart shows what happens when a customer uses a credit card to pay forgoods at a supermarket. Ten of the boxes are blank.

Using the items from the list, insert the ten missing statements using the appropriatenumber only. Each statement may be used once only.



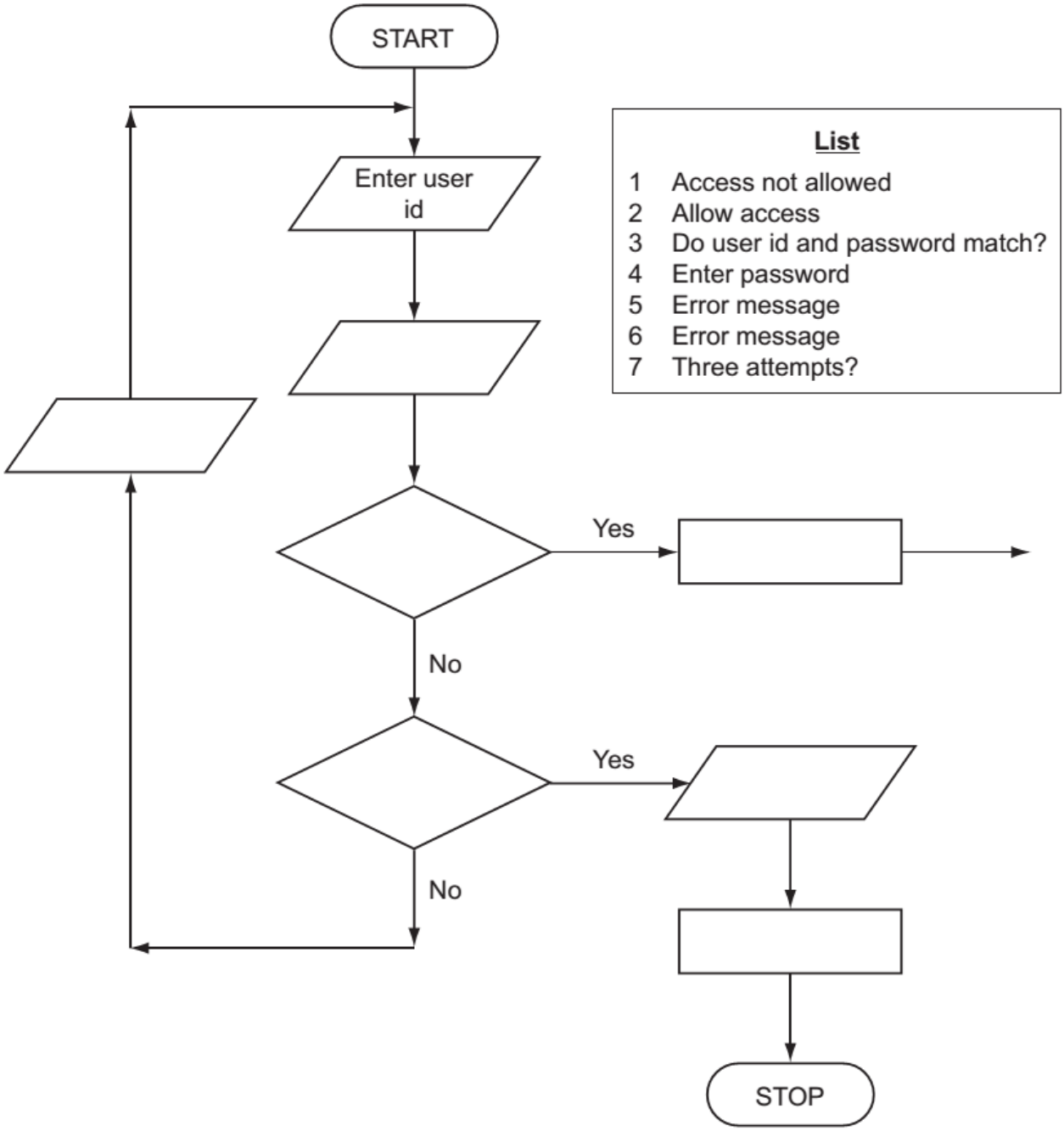
LIST OF STATEMENTS

1 are PIN details correct?
2 are there sufficient funds?
3 card removed
4 end transaction
5 is expiry date exceeded?
6 is card reported stolen?
7 money transfer takes place
8 print itemised bill
9 read credit card details
10 two computers communicate

Q5: Winter 2010. P11

(a) To log on to a computer, a user needs to type in a user id followed by a password; these should match up. Only three attempts are allowed.

The flowchart below shows the log on procedure. Several boxes have been left blank. Complete the flowchart using items from the list.

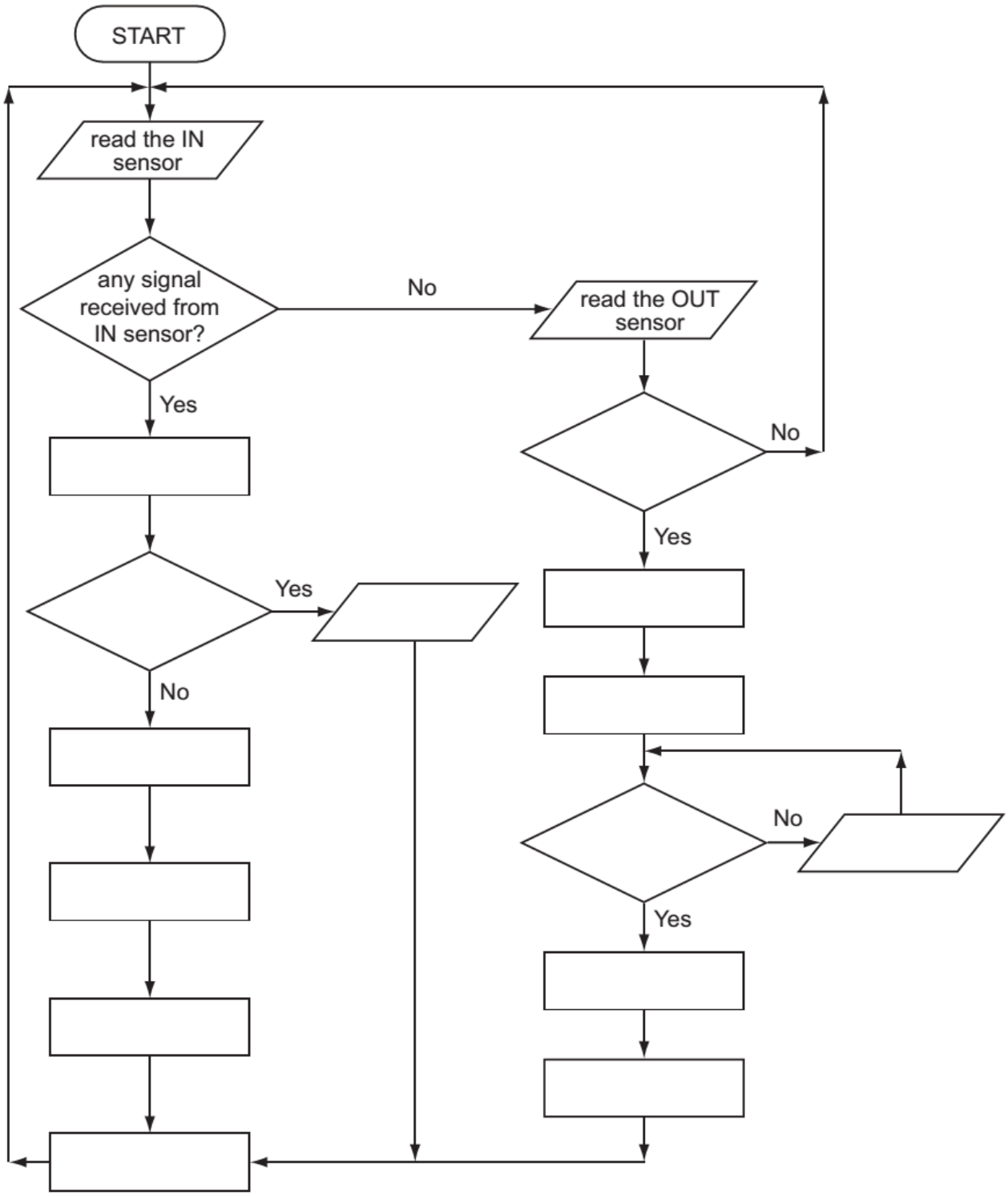


Q 6: Summer 2012. P11

A car park uses sensors and a microprocessor to monitor cars leaving and entering. The car park is open 24 hours every day. The park fee is $10 per day.

The following flowchart shows how the IN and OUT barriers are controlled. Some of the statements are missing.

Using item numbers only, insert the correct item number into the flow chart from the item list.



List of statements

Number Description

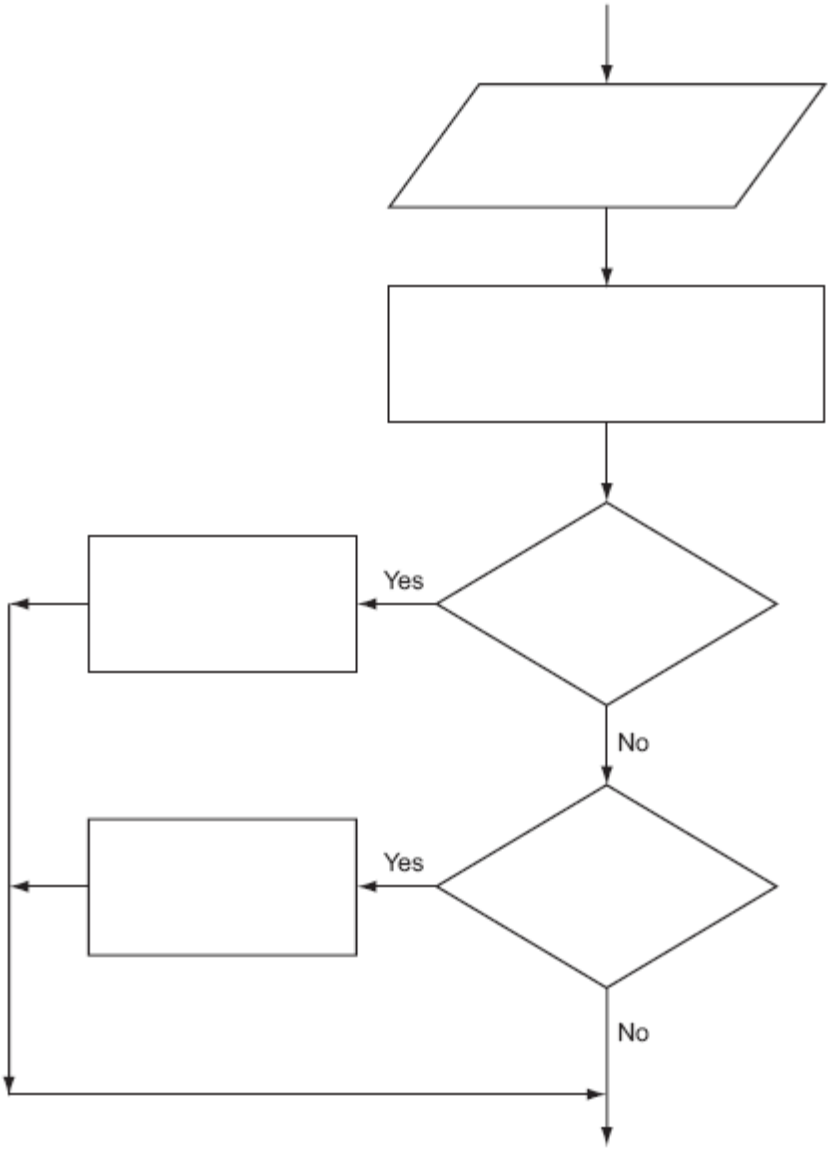
1 activate motor to raise IN barrier
2 activate motor to raise OUT barrier
3 any signal received from OUT sensor?
4 decrease number of cars in car park by 1
5 increase number of cars in car park by 1
6 is car park full?
7 is the car park fee paid?
8 OUTPUT “car park full”
9 OUTPUT “please pay car park fee at pay machine”
10 use ADC to convert IN sensor signal to digital
11 use ADC to convert OUT sensor signal to digital
12 use DAC to convert computer signal to analogue signal to operate IN barrier
13 use DAC to convert computer signal to analogue signal to operate OUT barrier
14 wait 30 seconds and then close barrier [6]

Q7:

(b) The computer also checks on humidity levels (using humidity sensors) which must be between the values of 40 and 90.

If humidity is too low, water is sprayed into the air.
If humidity is too high, fresh air is allowed to enter.

Write the necessary commands in the following flowchart section to show how the humidity levels are controlled:



[4]

Q8:Summer 2013 P11

A customer uses Internet banking. To gain access to their account they need:

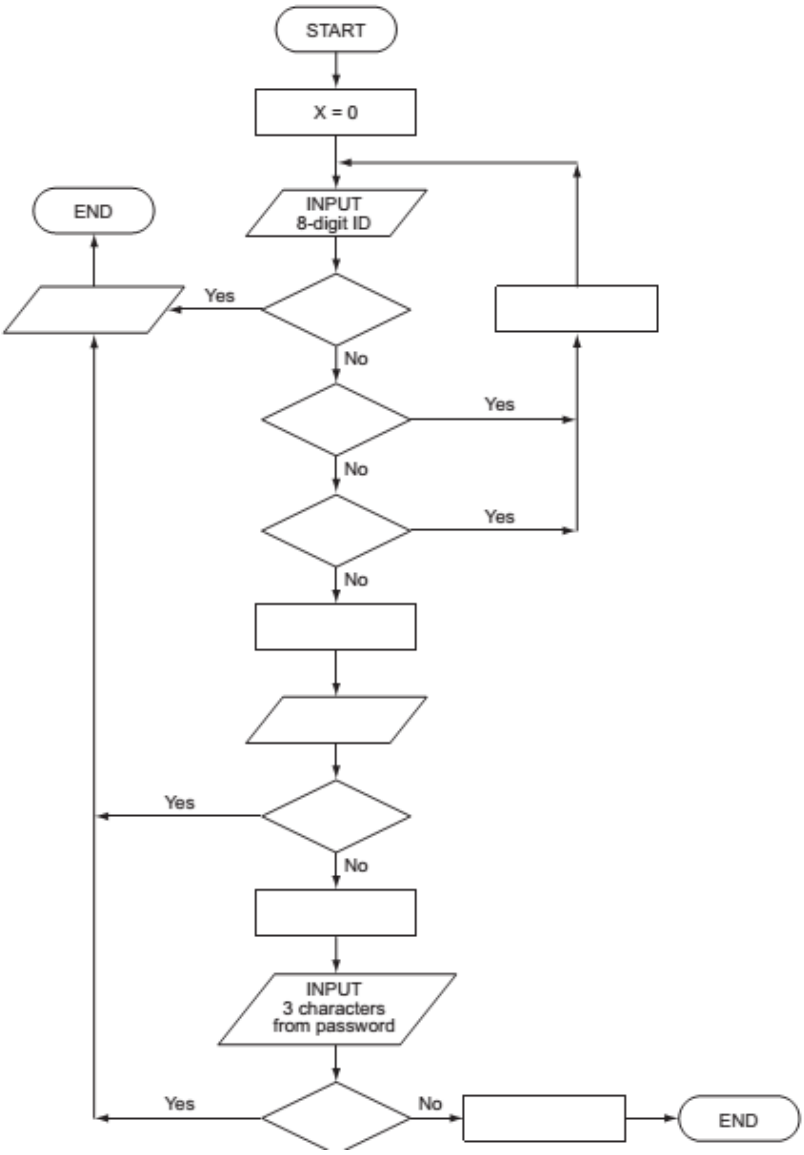
- an 8-digit ID
- a 4-digit PIN
- a 10-character password

They will be asked to type in their ID, then 3 digits from their PIN and finally 3 characters from their password. Three attempts at the ID are allowed, but only one attempt at the PIN and at the password.

The flowchart on the next page shows the access process described above. However, most of the stages have been omitted.

Complete the flowchart, using item number only, from the list of items given.

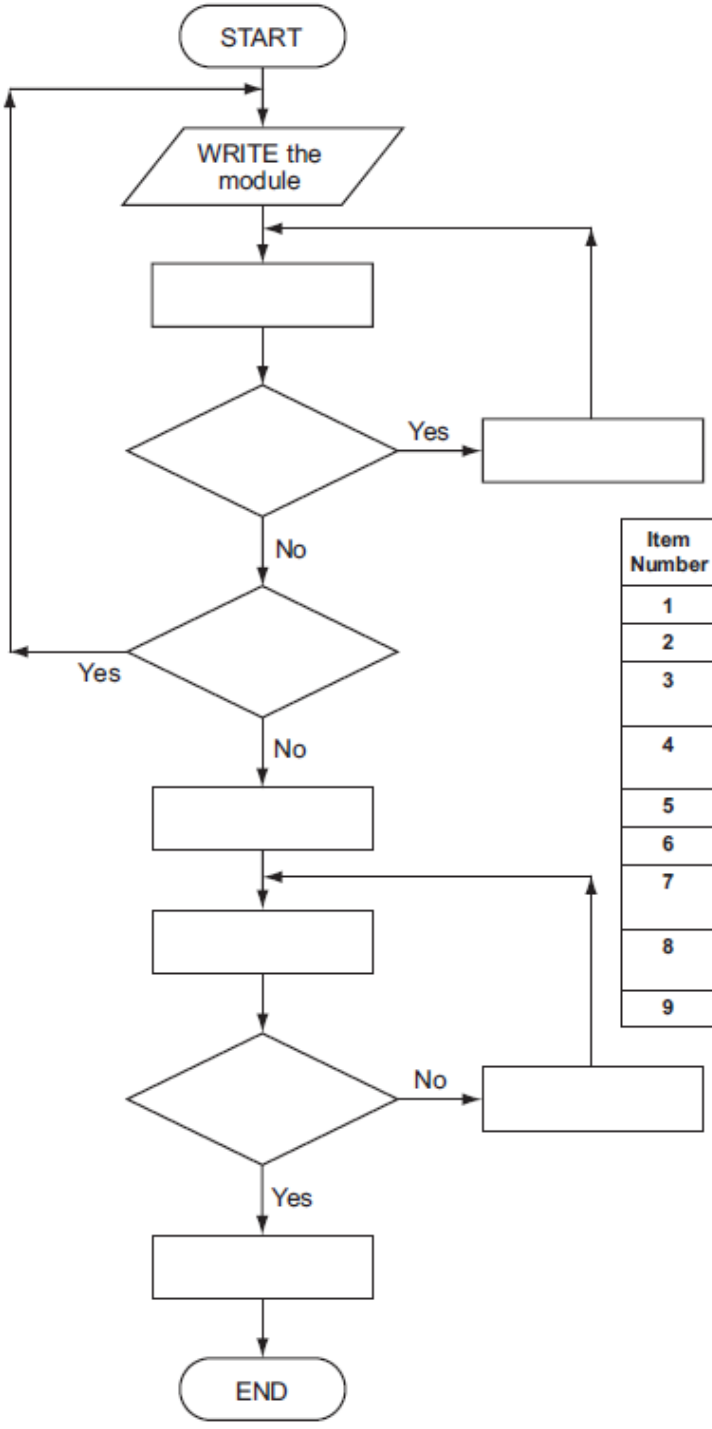
Item number	Item description
1	access to account allowed
2	are any characters in the password incorrect?
3	are any digits in the ID incorrect?
4	are any digits in the PIN incorrect?
5	generate three random digits from the PIN
6	generate three random characters from the password
7	input the required three digits from the PIN
8	is number of digits < 8 or number of digits > 8?
9	is X > 2?
10	output "access to account denied"
11	X = X + 1



Q9: Summer 2013. P12

16 A large word processor is being developed by first writing a series of modules. These are then put together to form the final word processor. Testing is done on each module and on the final word processor. The following flowchart shows how this word processor is developed. Several of the stages have been omitted.

Complete the flowchart, **using item number only**, from the list of items given.



ITEM LIST

Item Number	Item Description
1	any errors in the module?
2	any more modules to write and test?
3	combine all modules to form final word processor
4	does final word processor give expected results?
5	modify final word processor
6	modify the module
7	test module using data with known outcomes
8	test final word processor using data with known outcomes
9	write user documentation

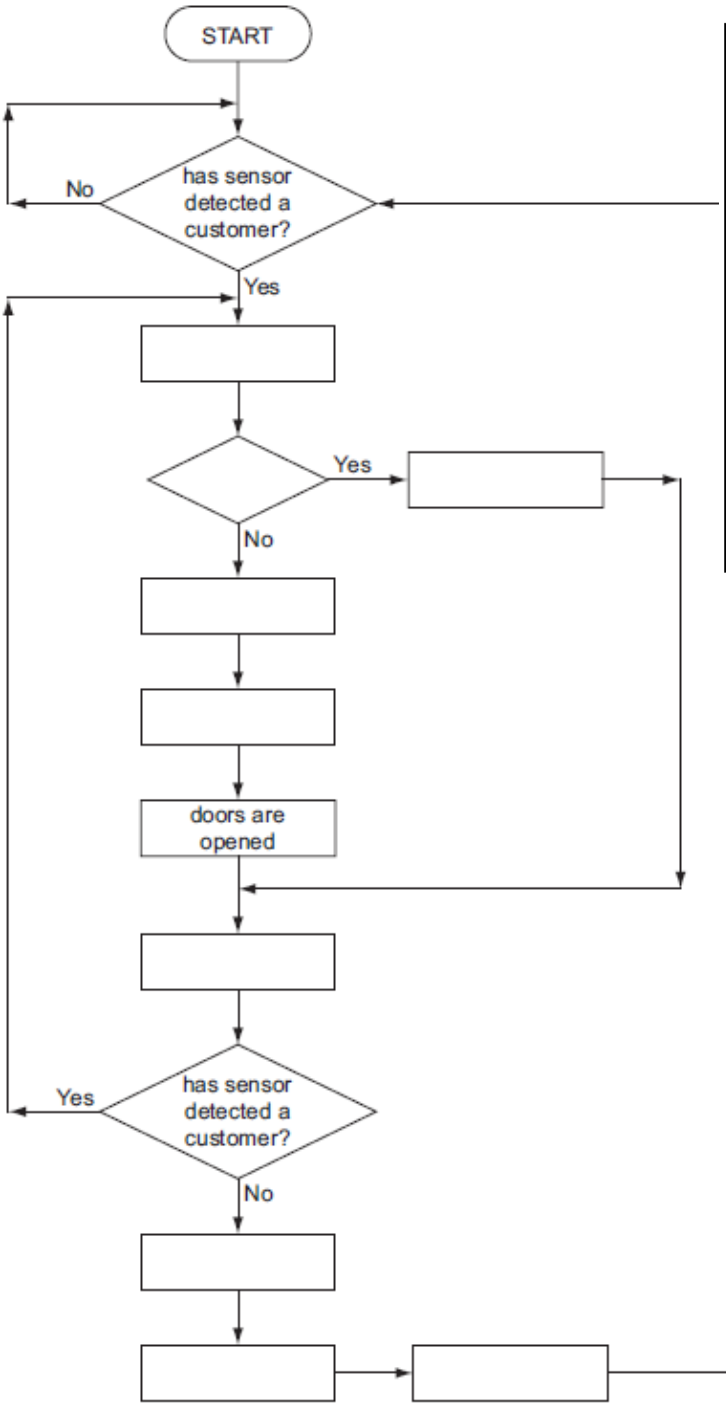
[5]

Q10: Winter 2013. P13

A microprocessor controls the opening and closing of automatic doors to a supermarket.Customers are detected using pressure sensors.

The flowchart on the next page shows how the sensors and microprocessor interacttocontrol the opening and closing of the doors. However, several of the stages in the processhave been missed out.

Using item number only, complete the flowchart using items from the following list:

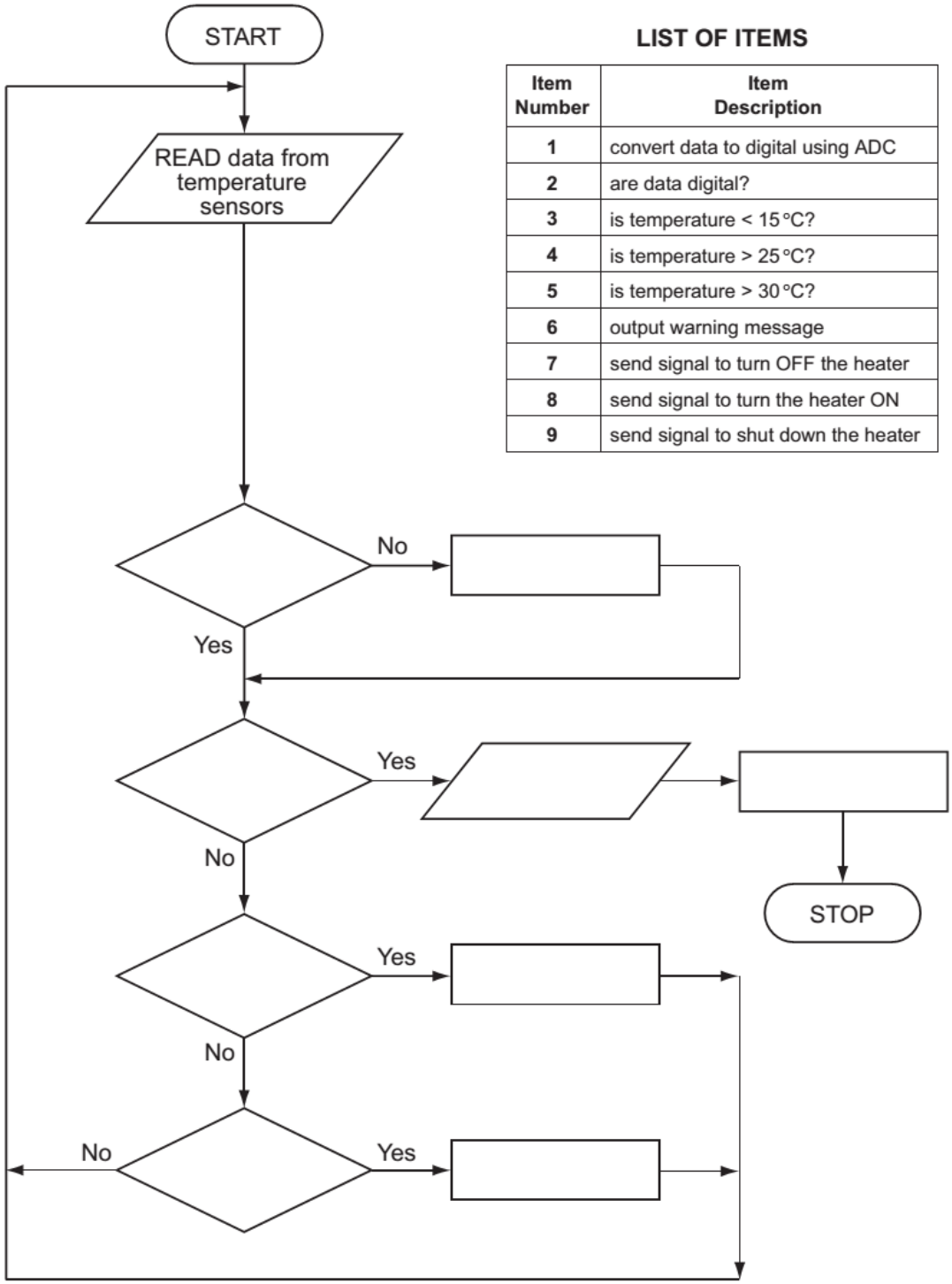


Item number	Item description
1	ADC changes analogue signal into digital signal
2	are the doors already open?
3	DAC changes digital signal into analogue signal
4	DAC changes digital signal into analogue signal
5	doors are closed
6	microprocessor sends signal
7	microprocessor sends signal
8	no action is required
9	wait for 10 seconds

[5]

Q11 :**Summer 2014. P11**

A heating system is being controlled by sensors and a computer. The temperature must be keptbetween 15°C and 25°C. If 30°C is exceeded a warning message is generated and the systemshutsdown.A flowchart of the process is shown below. Some of the items are missing.Complete the flowchart, using item number only, from the list of items given.



LIST OF ITEMS

Item Number	Item Description
1	convert data to digital using ADC
2	are data digital?
3	is temperature < 15 °C?
4	is temperature > 25 °C?
5	is temperature > 30 °C?
6	output warning message
7	send signal to turn OFF the heater
8	send signal to turn the heater ON
9	send signal to shut down the heater

Q12: Summer 2014

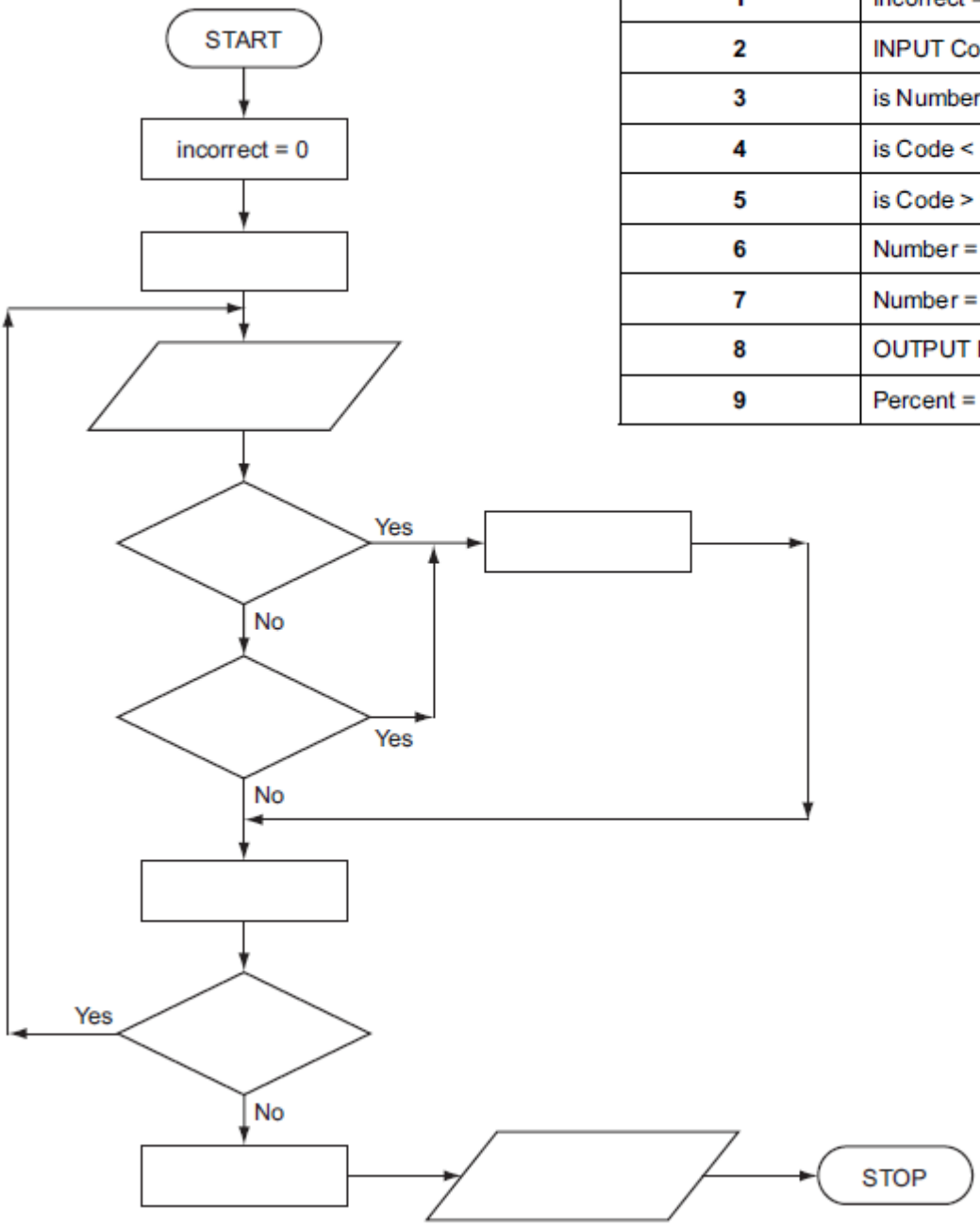
An algorithm has been written to check that code numbers are valid on input. They must be in therange 1000 to 9999.

Five hundred codes are being entered and the percentage of entered codes which are incorrectis output.

There is a flowchart on the opposite page. It has some statements missing.

Complete the flowchart. Use statement numbers only, chosen from the list below.

statement number	statement
1	Incorrect = Incorrect + 1
2	INPUT Code
3	is Number <= 500 ?
4	is Code < 1000 ?
5	is Code > 9999 ?
6	Number = 1
7	Number = Number + 1
8	OUTPUT Percent
9	Percent = Incorrect / 5



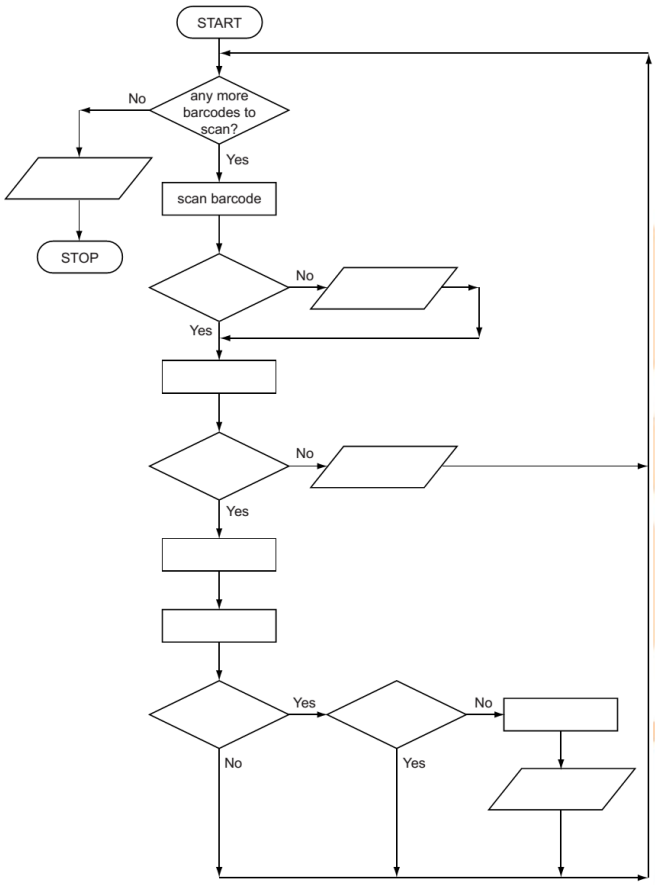
[4]

Q13: Winter 2014 P12

(a) The flowchart on the next page shows how barcodes are used in a supermarket to findproduct information and to produce orders for new stock automatically. Several statementsare missing from the flowchart.

Complete the flowchart, using item numbers **only** from the list below.

Item number	Description
1	is barcode found?
2	is barcode read?
3	is flag for this product = 1?
4	is number in stock <= re-order value?
5	key in the barcode manually
6	locate price and product information from file
7	output an error message
8	output order request for new stock
9	output receipt and itemised bill
10	reduce number in stock by 1 and write new value back to the record
11	search database for barcode
12	set flag for this product to 1



(b) Two devices used by the supermarket Point-Of-Sale (POS) terminal are a barcode reader and a keyboard.

Name **two** other input/output devices used at the POS and give a use for **each** device.

device 1 .....

use.....

.....

device 2 .....

use .....

.....

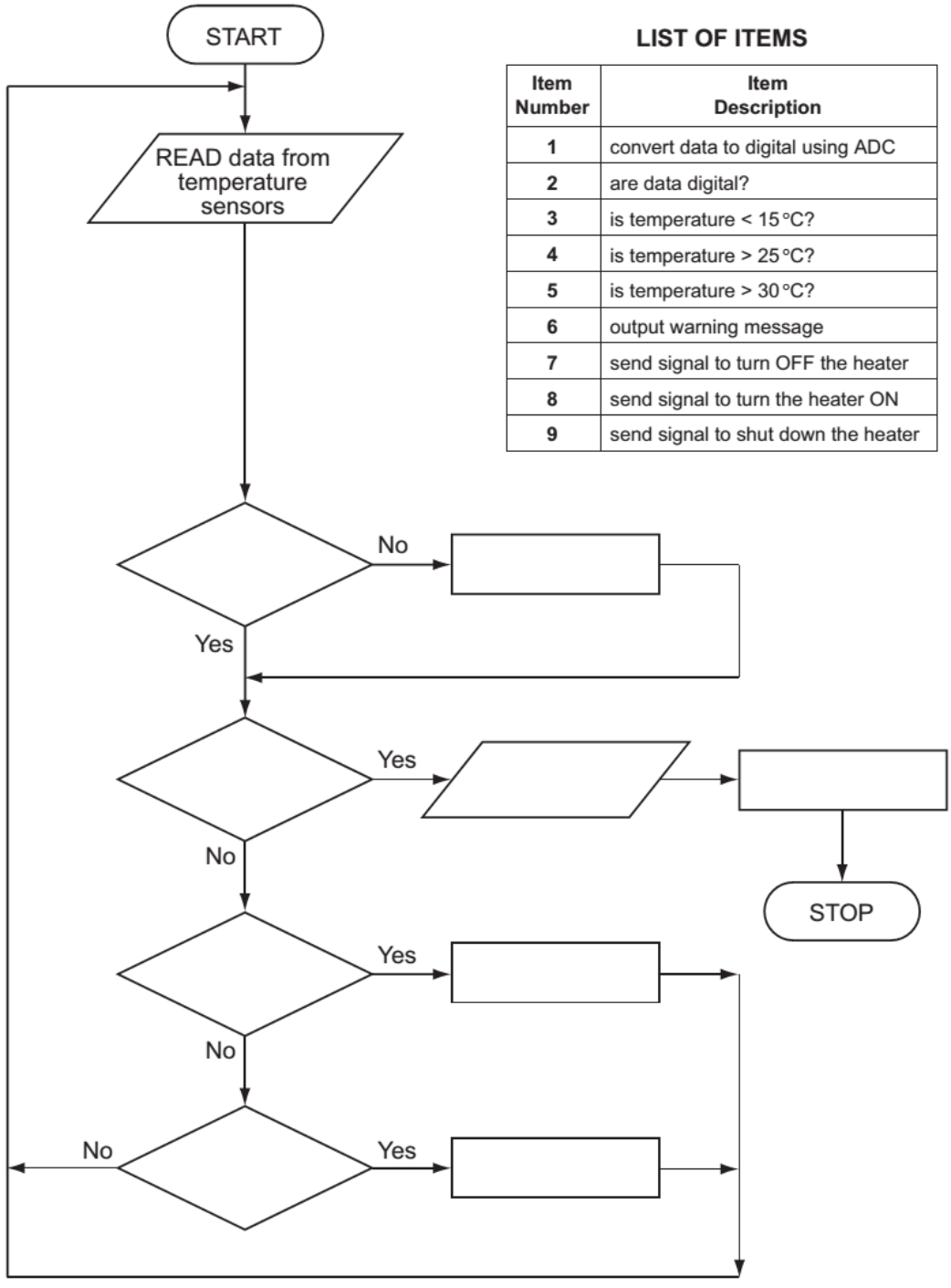
.....[4]

Q14: Summer 2014 P11

A heating system is being controlled by sensors and a computer. The temperature must be kept between 15°C and 25°C. If 30°C is exceeded a warning message is generated and the system shuts down.

A flowchart of the process is shown below. Some of the items are missing.

Complete the flowchart, using item number only, from the list of items given.



LIST OF ITEMS

Item Number	Item Description
1	convert data to digital using ADC
2	are data digital?
3	is temperature < 15 °C?
4	is temperature > 25 °C?
5	is temperature > 30 °C?
6	output warning message
7	send signal to turn OFF the heater
8	send signal to turn the heater ON
9	send signal to shut down the heater

Q15:

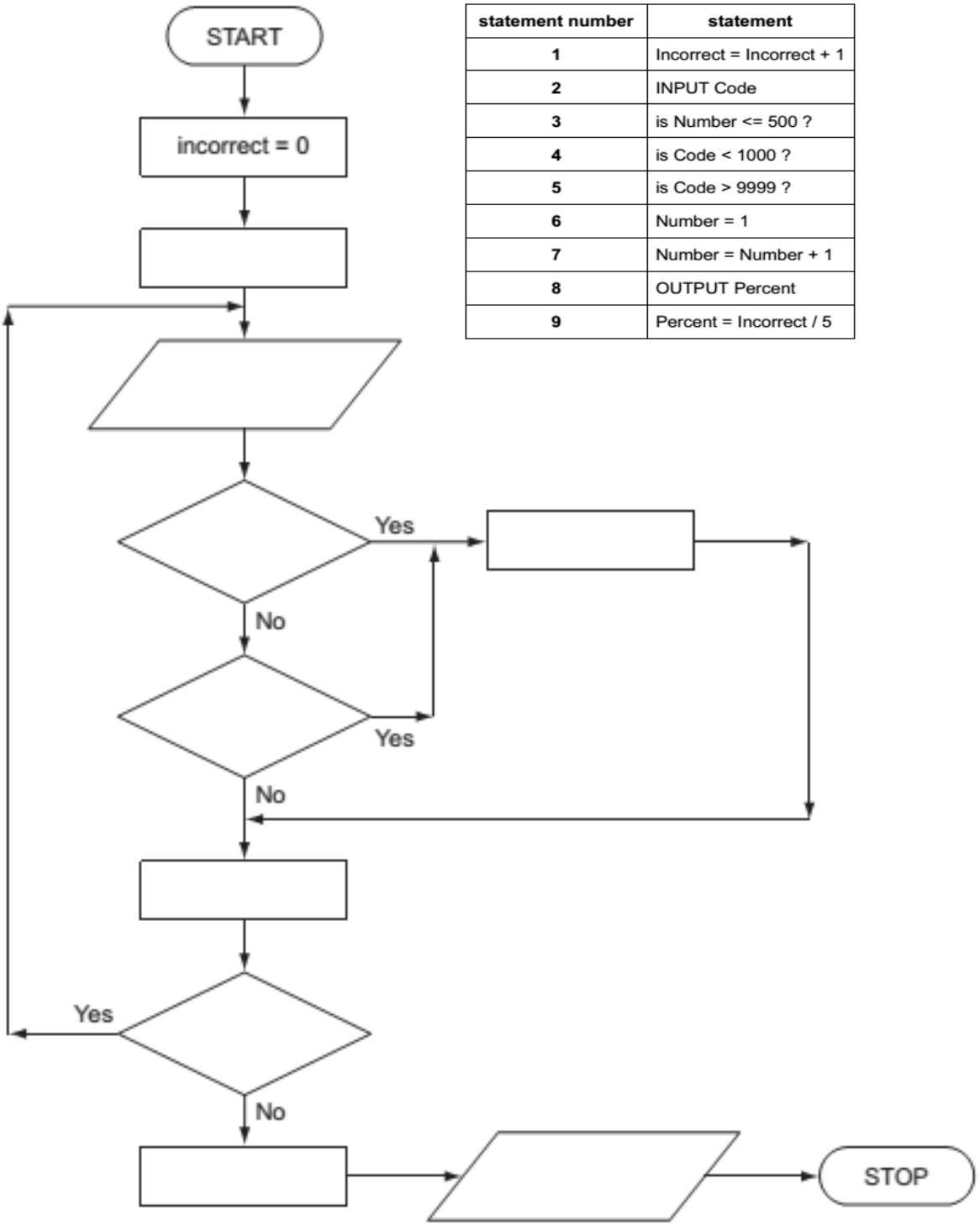
An algorithm has been written to check that code numbers are valid on input. They must be in the range 1000 to 9999.

Five hundred codes are being entered and the percentage of entered codes which are incorrect is output.

There is a flowchart on the opposite page. It has some statements missing.

Complete the flowchart. Use statement numbers only, chosen from the list below.

statement number	statement
1	Incorrect = Incorrect + 1
2	INPUT Code
3	is Number <= 500 ?
4	is Code < 1000 ?
5	is Code > 9999 ?
6	Number = 1
7	Number = Number + 1
8	OUTPUT Percent
9	Percent = Incorrect / 5



Q16

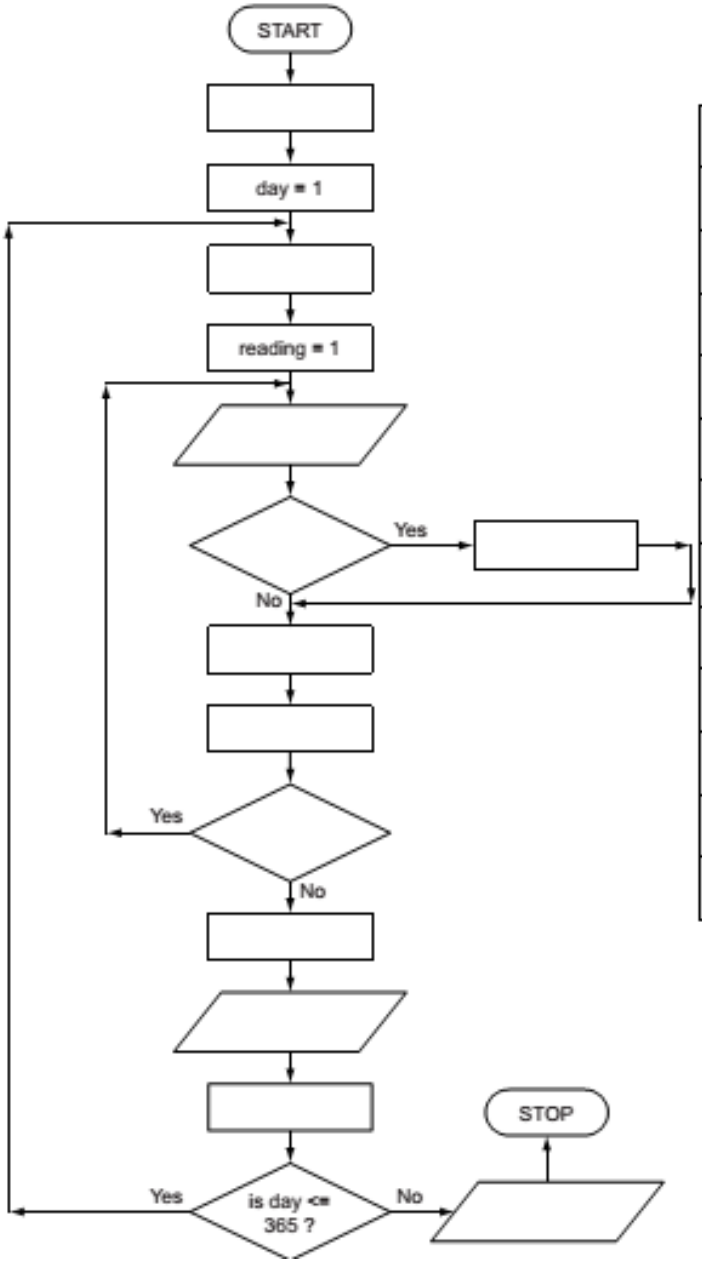
An algorithm has been written to input six temperatures for every day of the year (365 days). The outputs are:

• the average daily temperature for each day

• the highest recorded temperature for the whole year

The algorithm is in the form of a flowchart on the next page. However, several of the statements are missing.

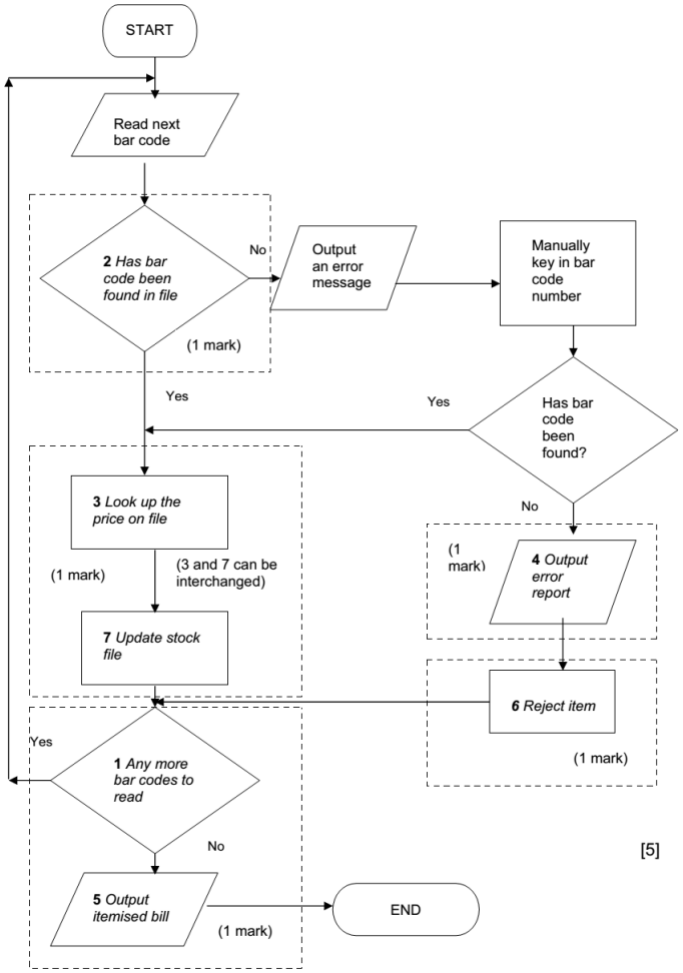
Using instruction number **only**, complete the flowchart using the following list of instructions:



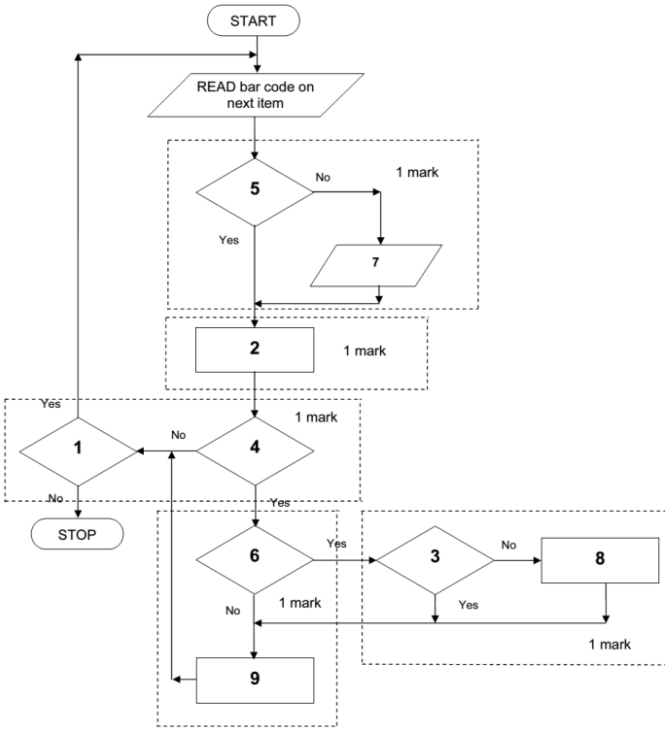
Instruction number	Instruction
1	average = total / 6
2	day = day + 1
3	high = -200
4	high = temperature
5	input temperature
6	is reading <= 6 ?
7	is temperature > high ?
8	output average
9	output high
10	reading = reading + 1
11	total = 0
12	total = total + temperature

Marking Scheme

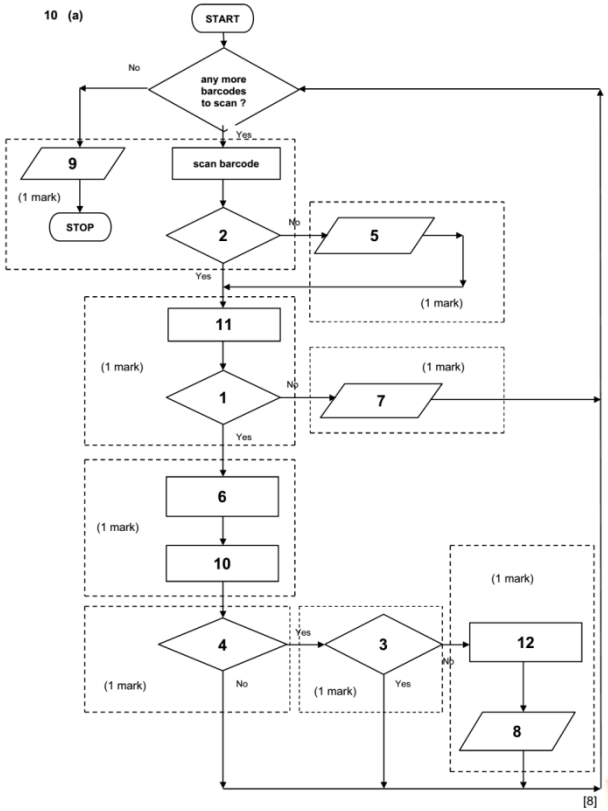
Q1)



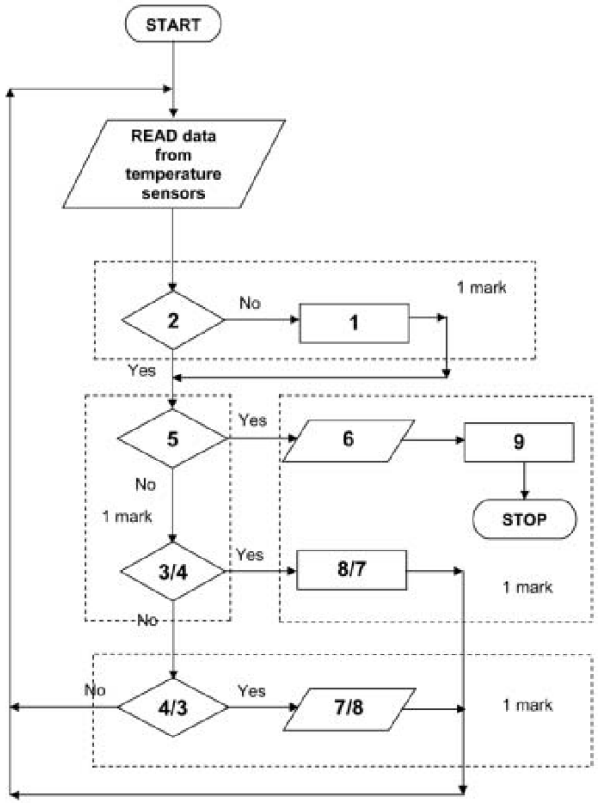
[5]



10 (a)



Q13



Q14) (3 and 8 AND 4 and 7 MUST be marked in PAIRS)
(accept the phrases)

2.2 Programming

Visual Basic .NET (VB.NET) is an object-oriented computer programming language implemented on the .NET Framework. Although it is an evolution of classic Visual Basic language, it is not backwards-compatible with VB6, and any code written in the old version does not compile under VB.NET. Like all other .NET languages, VB.NET has complete support for object-oriented concepts. Everything in VB.NET is an object, including all of the primitive types (Short, Integer, Long, String, Boolean, etc.) and user defined types, events, and even assemblies. All objects inherit from the base class Object.

VB.NET is implemented of Microsoft's .NET framework. Therefore it has full access all the libraries in the .Net Framework. It's also possible to run VB.NET programs on Mono, the open-source alternative to .NET, not only under Windows, but even Linux or Mac OSX.

The following reasons make VB.Net a widely used professional language:

- Modern, general purpose.
- Object oriented.
- Component oriented.
- Easy to learn.
- Structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.
- Part of .Net Framework.
- Strong Programming Features VB.Net

VB.Net has numerous strong programming features that make it endearing to multitude of programmers worldwide. Let us mention some of these features:

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy to use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading

VB.NET – ENVIRONMENT

Integrated Development Environment (IDE) For VB.Net

Microsoft provides the following development tools for VB.Net programming:

- Visual Studio 2010 (VS)
- Visual Basic 2010 Express (VBE)
- Visual Web Developer

The last two are free. Using these tools you can write all kinds of VB.Net programs from simple command-line applications to more complex applications. Visual Basic Express and Visual Web Developer Express edition are trimmed down versions of Visual Studio and has the same look and feel. They retain most features of Visual Studio. In this tutorial, we have used Visual Basic 2010 Express and Visual Web Developer

Download and Setup

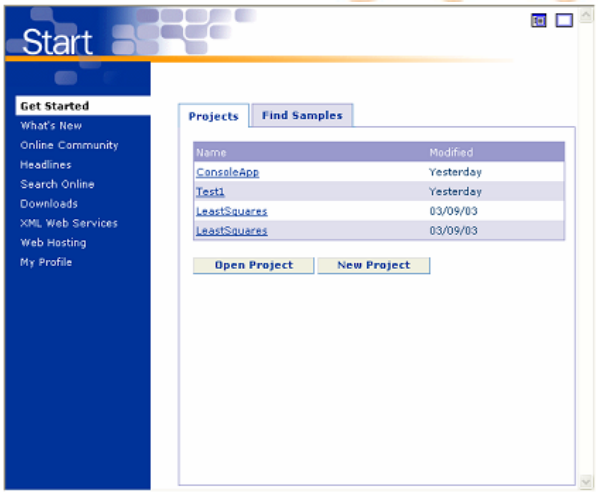
Download VB Express and install it on your computer at home.
http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express

If you don't run Windows as your operating system you can try using Mono
http://www.go-mono.com/mono-downloads/download.html

Console Application

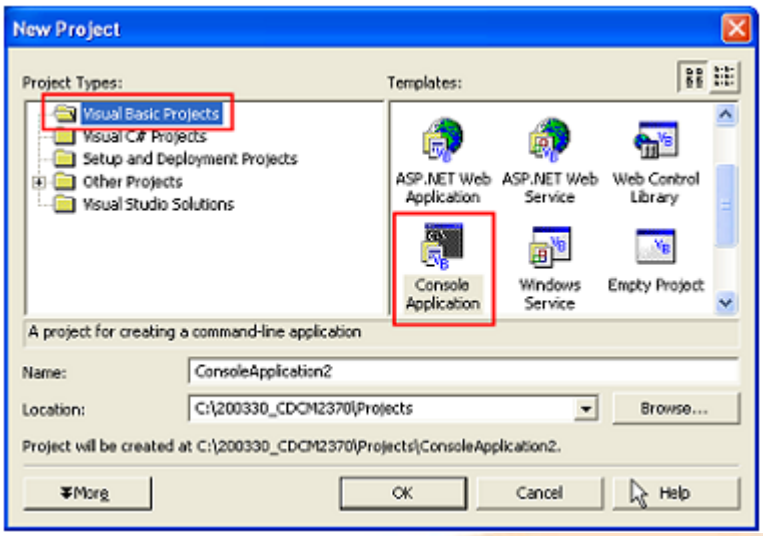
A VB.NET console application uses a command line window for its user interface. Because there is no opportunity for the user to provide inputs other than those asked for by the program, a console application is not "event driven". The actions performed in the program must follow in a linear fashion, and are completely defined by the program, and thus the programmer. This "procedural" programming is limiting, but makes a good introduction since the complexity of the user interface is eliminated.

1. Start Microsoft Visual Studio.NET. Note that it is not listed as Visual Basic.NET. Visual Studio.NET supports other programming languages other than VB. A Start Screen is displayed to help you open new or existing VB projects.

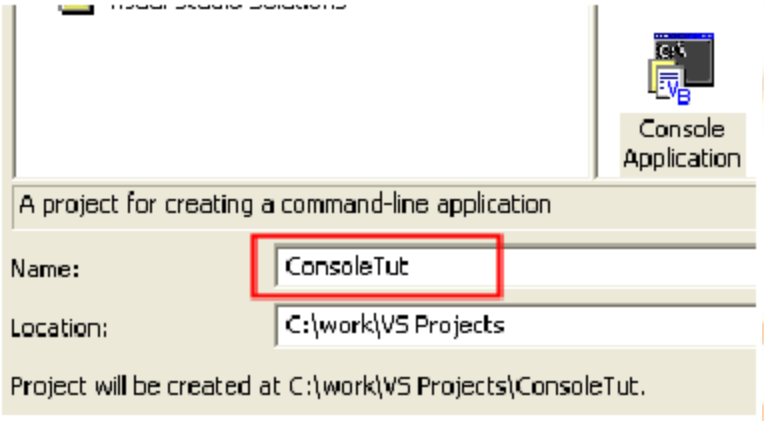


2. Click the New Project button.
3. The New Project dialog box is shown. You select the type of project and name it here.

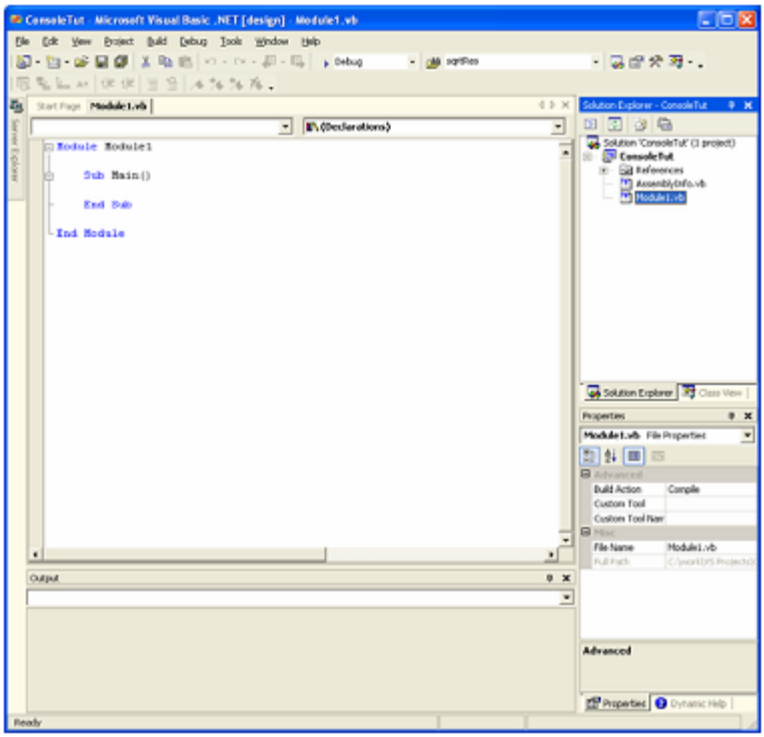
4. Ensure that the Visual Basic projects folder is opened, and then scroll to and click Console Application from the list of templates.



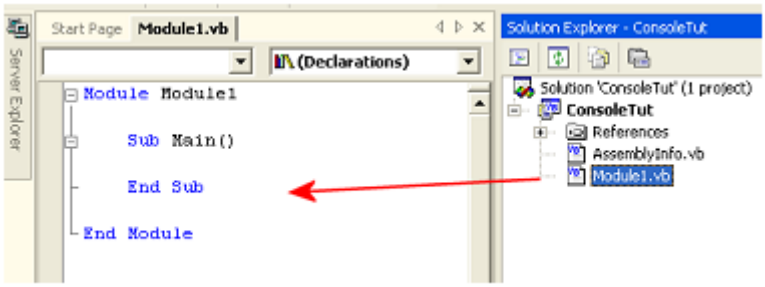
5. Enter ConsoleTut as the name of the project. All the files associated with the project will be stored in a folder as indicated in the Location text box (note this so you can find it later). Click OK to create the project.



6. The VS (Visual Studio) IDE (Integrated Development Environment) is where you build your program. You can actually create any VB program in Notepad, but the IDE provides a large number of tools that greatly simplifies the process of creating a Windows program.



7. For this tutorial, you can ignore most of the windows and tools in the IDE. The Solution Explorer in the top right corner of the IDE shows the files associated with the project. The console application template provides the required files. The Module1.vb file is the active file, and its contents are shown in the editor window.



8. The code contained within the Module1.vb file is listed between the two statements:

```
Module Module1

End Module
```

The module currently contains a single subroutine (don't worry too much about terms just yet) called Main. The code inside the Main subroutine is automatically "run" when the program starts. You will add all the code for this tutorial in between the Sub Main() and End Sub statements.

Tools: (Console.ReadLine for Input, Console.WriteLine for Output)

The console application simplifies input and output in that the user interface is essentially text based. The user types in the input, and the output is posted to the same text window. This text window is referenced in your program code by using the keyword Console. Three "actions" associated with the Console object enable you to receive input (ReadLine) and output (Write and WriteLine) text characters to the console window.

Basic data types

In order for a computer system to process and store data effectively, different kinds of data are formally given different types. This enables:

- data to be stored in an appropriate way, for example, as numbers or characters
- data to be manipulated effectively, for example numbers with mathematical operators and characters with concatenation
- automatic validation in some cases.

INTEGER

An **INTEGER** is a positive or negative whole number that can be used with mathematical operators.

SINGLE

A **Single** is a positive or negative number with a fractional part. Single numbers can be used with mathematical operators. (In pseudocode REAL data type is used for numbers with fractional part)

CHAR

A variable or constant of type **CHAR** is a single character.
Gender = 'F'

STRING

A variable or constant of type **STRING** is several characters in length. Strings vary in length and may even have no characters: an empty string. The characters can be letters and/or digits and/or any other printable symbol.

BOOLEAN

A **BOOLEAN** variable can have only two values: TRUE or FALSE.

Variables:

A variable is a named location in the computer's memory that stores a certain type of data (character, string of characters, integer, real number, and so on). The variable is identified in the program by a unique name.
The values stored in any variable are changed during execution of program.
By storing each of the inputs in memory (in a variable), you can use them again and again within the program without requiring that the user re-input them.

Variable Declaration:

Before variables can be used to store data, they must be "Declared". These reserves the appropriate amount of memory to store the value assigned to the variable.
In the editor window, below Sub Main() but above End Sub, type:

```
DIM Name As String
DIM Weight1, Weight2, Weight_Difference  As Single
DIM Count As Integer
```

Constant:

A constant is a memory location in which stored values remain unchanged during execution of program.

Constant Declaration and Initialization:

Before constants can be used, they must be declared and initialized with permanent values. Declaration and assignment of permanent value is done in the same line like:

```
ConstTaxRate as single=0.15
ConstMaxMarksas single=100
```

Array:

Data stored in a computer is stored at any location in memory that the computer decides to use, which means that similar pieces of data can be scattered all over memory. This, in itself, doesn't matter to the user, except that to find each piece of data it has to be referred to by a name (known as a "variable name"). For example, if we want to store the 20 names of students in a group then each location would have to be given a different variable name. The first, "Abdullah", might be stored in location Name1, the second, "Rumaisa", might be stored inName2 , the third, "Rashid", could be stored in Name3.Creating the 20 different variable names is possible but these names are separate (as far as the computer is concerned) and do not relate to each other in any way. It would be far more sensible to force the computer to store them all together using the same variable name. However, this doesn't let me identify individual names. If I call the first one, Name(1) , the second one Name(2) and so on, it is obvious that they are all people's names and that they are distinguishable by their position in the list. A list like this is called an **array.**

Each element in the array is identified using its **subscript** or **index number**. The largest and smallest index numbers are called the *upper bound* and *lower bound* of the array.

	Name
1	Abdullah
2	Rumaisa
3	Rashid
4	Muzna
5	Laiba
6	Patel
7	Smith
19	Mani
20	Ghayas

Declaring an array

It is important declare the arrays before assigning values in it so that program can reserve that amount of space in its memory; otherwise, there may not be enough space when the program uses the data.

Declaration consists of telling the computer program:

- the identifier name of the array
- the sort of data that is going to be stored in the array, i.e. its data type
- Howmany items of data are going to be stored, so that it knows how much space to reserve.

Different programming languages have different statements for initialising the array but they all do the same thing. In Visual Basic, the statement is:

Dim Name(20) As String

This Dim statement declares:

- the identifier name: Name
- the upper bound: 20
- the data type: String.

The upper bound of 20 specifies that there can be a maximum of 21 data items, since Visual Basic starts with a subscript of zero. We do not have to fill the array; the upper bound of 20 indicates the maximum size.

The array that has been described in one dimension array so far is really only a list of single data items. It is possible to have an array which can be visualised as a two-dimensional table with rows and columns and a data value in each cell.

Reading data into an array

To assign data values to the elements of the array, we do this with assignment statements such as:

Name(6) = “Patel”

This places the string “Allan” at index position 6 in the array.

Similarly, the following statement places the string “Rashid” at index position 3 in the array.

Name(19) = “Mani”

First program

Start a new Project and select Console Application. Type the following code:

```
module module1
    sub main()
        console. writeline("In the name of Allah”)
        Console.writeline(“Hello World!")
        console. readKey()
    end sub
end module
```

(Press F5 to run the code)

Tasks

0.1 – Write a program that displays the message “Asslam-o-Alaikum”

0.2 – Write a program that displays the message “My name is Muhammad and I live in Brussels” (replace Dave and Brussels with your own information)

0.3 – Write a program that displays the lyrics to your national anthem. Each line of the song should be on a fresh line.

Example Program 1 - Assignment - *Integer, byte, real, Boolean, character, string, date/time.*

```
Module Module1
    Sub Main()
        Dim theNumber As Integer
        Dim theWord As String

        theWord = "Bird"
        theNumber = 9

        Console.WriteLine(theWord& " is the word")
        Console.WriteLine("And the number is " &theNumber)

        Console.ReadKey()

        theWord = "Cat"
        Console.WriteLine("Enter a number>")
        theNumber = Int(Console.ReadLine())
        Console.WriteLine("Now " &theWord& " is the word and the number is " &
        theNumber)

        Console.ReadKey()
```

```
        End Sub
End Module
```

Example Program 2 - Arithmetic - +, –, /, x, *DIV, MOD*

```
Module Module1
    Sub Main()
        Dim number1, number2, total As Integer

        Console.WriteLine("Enter first number")
        number1 = Int(Console.ReadLine())
        Console.WriteLine("Enter second number")
        number2 = Int(Console.ReadLine())
        total = number1 + number2
        Console.WriteLine("The total is " & total)

        Console.ReadKey()
    End Sub
End Module
```

Tasks
3.1) Write a program that divides a number entered by the user by 2
3.2) Write a program that displays the 7 times table
3.3) Write a program that displays any times table the user requests

Example Program 3 – Selection

```
Dim intInput As Integer

System.Console.WriteLine("Enter an integer…")
intInput = Val(System. Console.ReadLine())
If intInput = 1 Then
    System.Console.WriteLine("Thank you.")
ElseIfintInput = 2 Then
    System.Console.WriteLine("That's fine.")
ElseIfintInput = 3 Then
    System.Console.WriteLine("Too big.")
Else
    System.Console.WriteLine("Not a number I know.")
End If
Console.ReadKey()
```

Tasks
1. Write a program to ask the user what 24+9 is. Say “Excellent” if they get it right.
2. Write a program to ask the user “how many in a baker’s dozen?” and say “mostexcellent” if they get it right.
3. Write a program to ask the user to enter their age. If their age is under 18 then say“Sorry, this film is not for you”.

4. Write a program to ask the user for two numbers. Compare the first with the second and then print out one of three messages. Either the numbers are equal, the first isbigger, or the second is bigger. You will need more than one IF to solve this one.
5. Write a program which asks the user to enter their password. If they enter the word“PASSWORD” then display the message “Welcome to the treasure”, otherwisedisplay a message which says “go away, it’s all mine”.
6. Write a program which asks the user to enter a number between 1 and 10. If thenumber entered is out with this range then display a message “Sorry…out of range”.

Example Program 4 - Relational operators - =, <, >, <>, <=, >=

```
Module Module1
    Sub Main()
        Dim age As Integer
        Console.WriteLine("What is your age?")
        age = Int(Console.ReadLine())
        If age > 16 Then
            Console.WriteLine("You can drive!")
        Else
            Console.WriteLine("You are too young to drive")
        End If

        Console.ReadKey()
    End Sub
End Module
```

The symbols we can use to test for conditions are as follows:

<	Less than
<=	Less Than or Equal To
>	Greater than
>=	Greater Than or Equal To
==	IS Equal To
!= or <>	Not Equal To

Example Program 5 - Boolean operators - *NOT, AND, OR*

```
Module Module1
    Sub Main()
        Dim age, points As Integer
        Console.WriteLine("What is your age?")
        age = Int(Console.ReadLine())
        Console.WriteLine("How many points do you have on your licence?")
        points = Int(Console.ReadLine())
        If age > 16 And points < 9 Then
            Console.WriteLine("You can drive!")
        Else
            Console.WriteLine("You are not eligible for a driving licence")
        End If

        Console.ReadKey()
```

```
        End Sub
End Module
```

Example Program 6 - Built-in functions- *Arithmetic functions: round, truncation.*

```
Dim num As Double
Dim rounded As Integer
Dim squarert As Double
Dim trunc As Integer

Console.Write("Enter a real number")
num = Console.ReadLine()
rounded = Math.Round(num)
squarert = Math.Sqrt(num)
Console.WriteLine("round: " & rounded &vbNewLine& "Square Root: " &squarert)
trunc = Math.Truncate(num)
Console.WriteLine("The number truncated is " &trunc)
Console.WriteLine("This is not always the same as rounded")

Console.ReadKey()
```

Tasks

1) Write a program that asks for 5 numbers, calculates the mean average and then rounds it down. Display the result on screen.

Example Program 7 - String handling functions *length, position, substring, concatenation*.

```
Dim theString As String
theString = "Hello Dave, you're my wife now!"
Console.WriteLine(theString)
Console.WriteLine(theString.Length) 'display the string's length
Console.WriteLine(theString.ToUpper) 'display the string in upper case
Console.WriteLine(theString.ToLower) 'display the string in lower case
Console.WriteLine(theString.Contains("Dave")) 'is Dave there?
Console.WriteLine(theString.IndexOf("D")) 'position of D
Console.WriteLine(theString.Substring(12)) 'displays the substring starting at
position 12
Dim newString As String
newString = "Speak to Dave! " &theString 'string concatenation
Console.WriteLine(newString)
Console.ReadKey() ' pause and wait so user can read output.
```

Tasks

Tasks

1. Write a program that checks a username against a stored value. How the user enters the username should NOT be case sensitive.
2. Adapt program 1 so that it also takes in a password. If the user enters spaces after the password the computer will trim them out automatically.
3. Write a program that will check a phone number is of the correct length.
4. Write a program that asks for a user's full name in one inputbox/textbox but then stores the first and second names in different variables.

Example Program 8 – Repetition

```
Module Module1
    Sub Main()
        Dim theNumber As Integer
        theNumber = 7
        'a loop
        For x = 1 To 10
            Console.WriteLine("7 x " & x & " = " & (7 * x))
        Next
        'the end of the loop
        Console.ReadKey() 'pause so user can see
    End Sub
End Module
```

Tasks

1. Write a program which asks for your name and then displays it 5 times on the screen.
2. Write a program to display the name of the town you live in 10 times.
3. Write a program to ask for a person's favourite CD and the artist. Both should be displayed on the same line 5 times.
4. Write a program to ask for a number and display its multiplication table 1 to 100
5. Write a program that asks the user for a number 5 times and adds them all up to give a total.

Example Program 9 - Constants

```
'a constant is a value that doesn't change
'using them greatly improves the readability of your code
'number constants
    ConstconPi = 3.14159265358979

    ConstconMaxPlanets As Integer = 9
    'string constants
    ConstconVersion = "07.10.A"
    ConstconCodeName = "Enigma"
    ' try assigning a new value to one of your constants
    'to make a constant available to the whole program "Public" should precede it
    'and it should be placed inside the module but outside the procedure
    'try it
    'there are string constants created by VB
    Console.WriteLine("Pi is" &conPi)
    Console.WriteLine("Pi is" &vbCr&conPi)
    Console.WriteLine("Pi is" &vbTab&conPi)
    Console.ReadKey()
```

Example Program 10 – 1 dimensional arrays

```
Dim countries(5) As String
Dim randomNum As Integer
countries(1) = "Scotland"
countries(2) = "Belgium"
countries(3) = "Netherlands"
countries(4) = "Germany"
countries(5) = "France"
Randomize()
randomNum = Int(Int((5 * Rnd()) + 1))
Console.WriteLine("You should go to " & countries(randomNum) & " on holiday.")
Console.ReadKey()
```

Tasks

1) Write a program which will set up an array to hold 50 numbers. Call the array numbers. Display
the array's contents across the screen. They should all be 0.
2) Create a program that stores an array of car records. At least 5 cars and 4 fields per record.
3) Create a program that stores an array of 5 people records. The information should be entered by
the user.
4) Adapt program 2 to now do a linear search for a certain car and display its details.

Example Program 11 - Validation

```
Dim mark As Integer
Do
        Console.WriteLine("Enter a mark between 0 and 10")
        mark = Val(Console.ReadLine())
        If (mark > 10) Or (mark < 0) Then Console.WriteLine("That was not a valid mark")
Loop Until (mark >= 0) And (mark <= 10) ' keeps going until a valid mark is entered
Console.WriteLine("Well done!")
Console.ReadKey()
```

Tasks

1) Write a program that validates a user is old enough to drive (older than 17, younger than 80)
2) Write a program that checks that a telephone number entered is long enough (string length)
3) Write a program that checks that both a username and password is correct before allowing you to proceed.

Example Program 12 – Read from a text file

```
Dim objStreamReader As IO. StreamReader
Dim strLine As String
'Pass the file path and the file name to the StreamReader constructor.
objStreamReader = New IO. StreamReader("H: \Diary.txt")
'Read the first line of text.
strLine = objStreamReader.ReadLine
'Continue to read until you reach the end of the file.
Do While Not strLineIs Nothing
'Write the line to the Console window.
Console.WriteLine(strLine)
'Read the next line.
strLine = objStreamReader.ReadLine
Loop
'Close the file.
objStreamReader.Close()
Console.ReadLine()
```

Tasks

1) Write a program that reads the students' names from a txt file and displays them on the screen

2) Write a program that reads 10 team names from a txt file and stores them in an array

3) Write a program that reads 5 song titles from a csv file and displays them on the screen

4) Write a program that reads 20 team names from a csv file into an array, then displays the array on screen

PRE-RELEASE MATERIAL **MAY/JUNE 2016**

In preparation for the examination candidates should attempt the following practical tasks by writing and testing a program(s)

The manager of a building materials delivery service needs a program to check the contents and weight of sacks to ensure that correct orders are made up for delivery. A price for the order will be calculated.

Write and test a program for the manager

- Yourprogram must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly.
- All variables, constants and other identifiers must have meaningful names.

TASK 1-check the contents and weight of a single sack

Each sack must obey the following rules to be accepted:

- Contain cement, gravel or sand, with a letter on the side for easy identification
  - C-cement
  - G-gravel
  - S-sand
- Sand and gravel must weigh over 49.9 and under 50.1 kilograms
- Cement must weigh over 24.9 and under 25.1 kilograms

Input and store the weight and contents for one sack. The contents must be checked and an incorrect sack rejected. The weight must be validated on entry and an overweight or underweight sack rejected.

Output the contents and weight of an accepted sack. If a sack is rejected, output the reason(s).

TASK 2- check a customer's order for delivery

Input and store the number of sacks for each type required for the order. Use TASK 1 to check the contents and weigh of each sack. Ensure that the delivery contains the correct number and type of sacks for the order.

Output the total weight of the order.

Output the number of sacks rejected from the order.

TASK 3- calculate the price for a customer's order

Extend TASK 2 to calculate a price for an order. Prices for the sacks are as follows:

- Regular price for each sack
  - Cement,$3
  - Gravel,$2
  - Sand,$2
- Discount price for a special pack containing 1 sack of cement,2 sacks of sand and 2 sacks of gravel,$10.

Calculate and output the regular price for the order. Check how many special packs are in the order. If a discount price applies then output the new price for the order and the amount saved.

TASK 1 – Check the contents and weight of a single sack

Each sack must obey the following rules to be accepted:

- contain cement, gravel or sand, with a letter on the side for easy identification
  - C - cement
  - G - gravel
  - S - sand
- sand or gravel must weigh over 49.9 and under 50.1 kilograms
- cement must weigh over 24.9 and under 25.1 kilograms

Input and store the weight and contents for one sack. The contents must be checked and an incorrect sack rejected. The weight must be validated on entry and an overweight or underweight sack rejected.

Output the contents and weight of an accepted sack. If a sack is rejected, output the reason(s).

PSEUDO CODE FOR TASK 1

#Prerelease2016

```
DECLARE Content: Char
DECLARE Weight: Real
PRINT“Enter content of sack “
READ Content
IF  Content =  “C”     THEN
        PRINT“weight of the sack “
        READ Weight
        CASE OFWeight
        <=24.9:         PRINT ”Sack is rejected due to underweight content”
        >=25.1:         PRINT ”Sack is rejected due to overweight content”
        OTHERWISE:          PRINT Content, Weight
        ENDCASE
ELSEIFContent =“S”OR  “G” THEN
        PRINT“weight of the sack “
        READ Weight
        CASE OFWeight
        <=49.9:         PRINT ”Sack is rejected due to underweight content”
        >=50.1:         PRINT ”Sack is rejected due to overweight content”
        OTHERWISE:          PRINT Content, Weight
        ENDCASE
ELSE
        PRINT ”Sack is rejected due to invalid content”
ENDIF
```

VB.Net CODE FOR TASK 1

```
Module Module1

Sub Main()
Dim Content AsChar
Dim Weight AsSingle

        Console.Write("Enter Content of sack ")
        Content = Console.ReadLine
SelectCase Content
Case"C", "c" : Console.Write("Enter weight of sack ")
               Weight = Console.ReadLine
SelectCase Weight
CaseIs<= 24.9 : Console.WriteLine("Rejected, due under weight sack")
CaseIs>= 25.1 : Console.WriteLine("Rejected, due over weight sack")
CaseElse : Console.WriteLine("Content  =  "& Content)
                       Console.WriteLine("Weight   =  "& Weight)
EndSelect

Case"G", "g", "S", "s" : Console.Write("Enter weight of sack ")
               Weight = Console.ReadLine
SelectCase Weight
CaseIs<= 49.9 : Console.WriteLine("Rejected, due under weight sack")
CaseIs>= 50.1 : Console.WriteLine("Rejected, due over weight sack")
CaseElse : Console.WriteLine("Content  =  "& Content)
                       Console.WriteLine("Weight   =  "& Weight)
EndSelect
CaseElse : Console.WriteLine("Rejected, due to invalid content")

EndSelect
        Console.ReadKey()

EndSub

EndModule
```

TASK 2 – Check a customer's order for delivery

Input and store the number of sacks of each type required for the order. Use TASK 1 to check the contents and weight of each sack. Ensure that the delivery contains the correct number and type of sacks for the order.

Output the total weight of the order.
Output the number of sacks rejected from the order.

PSEUDO CODE FOR TASK 2

```
DECLARE Weight, TWeight:Real

DECLARE Content:Char
DECLARE Count, COrder, SOrder, GOrder, TOrder, CountRejected:Integer
DECLARE CDeliver, SDeliver, GDeliver, TDeliver :Integer

CountRejected ← 0
CDeliver ← 0
SDeliver ← 0
GDeliver ← 0
TWeight  ← 0

PRINT"Enter order for cement "
READCOrder
PRINT"Enter order for gravel "
READGOrder
PRINT"Enter order for sand   "
READSOrder

TOrder = COrder + SOrder + GOrder

WHILE CDeliver < COrder OR SDeliver < SOrder OR GDeliver < GOrder DO
      PRINT"Enter content of sack to deliver"
READ Content
IF CDeliver < COrder AND Content = "C"THEN
            PRINT"Enter weight of sack of cement "
            READ Weight

            CASE OF Weight

            <=24.9, >=25.1:    CountRejected ← CountRejected + 1

                               PRINT"Rejected due to invalid weight"
            OTHERWISE:         CDeliver = CDeliver + 1
                  TWeight = TWeight + Weight
            ENDCASE

ELSEIF GDeliver < GOrder AND Content = "G"THEN
            PRINT"Enter weight of sack of gravel "
            READ Weight

            <=49.9, >=50.1:    CountRejected ← CountRejected + 1

                               PRINT"Rejected due to invalid weight "
            OTHERWISE:         SDeliver = SDeliver + 1
                  TWeight = TWeight + Weight
            ENDCASE
```

```
ELSEIF SDeliver < SOrder AND Content = "S"Then
            PRINT"Enter weight of sack of sand "
            READ Weight
            <=49.9, >=50.1:    CountRejected ← CountRejected + 1
                               PRINT"Rejected due to invalid weight"
            OTHERWISE:         GDeliver = GDeliver + 1
                  TWeight = TWeight + Weight
            ENDCASE
      ELSE
            CountRejected ← CountRejected + 1
            PRINT"Rejected due to completed order or invalid content "
      ENDIF
ENDWHILE
PRINT "Total Rejected " ,CountRejected
PRINT"Total weight delivered " , TWeight)
```

Task 2 (VB.NET Console)


```
      Dim Content AsChar
Dim Weight, TWeight AsSingle
Dim COrder, GOrder, SOrder AsInteger
Dim CDeliver, SDeliver, GDeliver AsInteger
Dim CountRejected AsInteger
        TWeight = 0
        CDeliver = 0
        GDeliver = 0
        SDeliver = 0
        CountRejected = 0
        Console.Write("Enter order for cement ")
        COrder = Console.ReadLine
        Console.Write("Enter order for gravel ")
        GOrder = Console.ReadLine
        Console.Write("Enter order for sand   ")
        SOrder = Console.ReadLine
While CDeliver < COrder Or GDeliver < GOrder Or SDeliver < SOrder
            Console.Write("Enter content of sack to deliver ")
            Content = Console.ReadLine
If (Content = "C"Or Content = "c") And CDeliver < COrder Then
                Console.Write("Enter weight of sack of cement ")
                Weight = Console.ReadLine
SelectCase Weight
CaseIs<= 24.9, Is>= 25.1
                        Console.WriteLine("Rejected due to invalid weight")
                        CountRejected = CountRejected + 1
CaseElse : TWeight = TWeight + Weight
                        CDeliver = CDeliver + 1
EndSelect
ElseIf (Content = "S"Or Content = "s") And SDeliver < SOrder Then
                Console.Write("Enter weight of sack of sand ")
                Weight = Console.ReadLine
SelectCase Weight
CaseIs<= 49.9, Is>= 50.1
                        Console.WriteLine("Rejected due to invalid weight")
                        CountRejected = CountRejected + 1
CaseElse : TWeight = TWeight + Weight
                        SDeliver = SDeliver + 1
EndSelect
ElseIf (Content = "G"Or Content = "g") And GDeliver < GOrder Then
                Console.Write("Enter weight of sack of gravel ")
                Weight = Console.ReadLine
SelectCase Weight
CaseIs<= 49.9, Is>= 50.1
                        Console.WriteLine("Rejected due to invalid weight")
                        CountRejected = CountRejected + 1
CaseElse : TWeight = TWeight + Weight
                        GDeliver = GDeliver + 1
EndSelect
Else
                Console.WriteLine("Rejected   due   to   invalid   content   of
completed order")
                CountRejected = CountRejected + 1
EndIf
EndWhile
        Console.WriteLine("Total weight of order= "& TWeight)
        Console.WriteLine("Total sacks rejected = "& CountRejected)
        Console.ReadKey()
```

TASK 3 – Calculate the price for a customer's order

Extend TASK 2 to calculate a price for an order. Prices for the sacks are as follows:

- regular price for each sack
  - cement, $3
  - gravel, $2
  - sand, $2
- discount price for a special pack containing 1 sack of cement, 2 sacks of sand and 2 sacks of gravel, $10

Calculate and output the regular price for the order. Check how many special packs are in the order. If a discount price applies then output the new price for the order and the amount saved.

PSEUDO CODE FOR TASK 3

```
//Declaration of constants & variables
CONSTANT CPrice=3
CONSTANT GPrice=2
CONSTANT SPrice=2
CONSTANT SpecialPrice=10
DECLARE RegularPrice, DiscountPrice, AmountSaved:Real
DECLARE C, G, S, SpecialPack:Integer

//Calculation of regular price
RegularPrice ← (COrder x CPrice) + (SOrder x SPrice) + (GOrder x GPrice)
PRINT "Total Price of order     = " , RegularPrice

//Initialization of temporary variables and SpecialPack
C ← COrder
G ← GOrder
S ← SOrder
SpecialPack ← 0

//Calculation of number of special packs in order
WHILE C>=1 AND S>=2 AND G>=2 DO
      SpecialPack ← SpecialPack + 1
      C ← C - 1
      S ← S - 2
      G ← G - 2
ENDWHILE

//Calculation of discount price & amount saved
IF SpecialPack>=1 THEN
      DiscountPrice ← (SpecialPack   * SpecialPrice) + (C * CPrice) + (S *
SPrice) + (G * GPrice)
      AmountSaved ← RegularPrice - DiscountPrice
PRINT "Discount Price of order  = " , DiscountPrice
PRINT"Amount Saved on discount = " , AmountSaved
ENDIF
```

Task 3 (VB.NET Console)


```
'Declaration of variables
Dim C, G, S, SpecialPack AsInteger
Dim RegularPrice, DiscountPrice, AmountSaved AsSingle

'Declaration of constants
Const CPrice = 3
Const GPrice = 2
Const SPrice = 2
Const SpecialPrice = 10


'initialization of special pack and temporary variables
        SpecialPack = 0
        C = COrder
        G = GOrder
        S = SOrder

'Calculation of regular price
        RegularPrice = (C * CPrice) + (G * GPrice) + (S * SPrice)
        Console.WriteLine("Regular price for order = "& RegularPrice)

'Calculation of special packs
While C >= 1 And G >= 2 And S >= 2
            SpecialPack = SpecialPack + 1
            C = C - 1
            G = G - 2
            S = S - 2
EndWhile

'Calculation of special price
If SpecialPack >= 1 Then
            DiscountPrice = (SpecialPack * SpecialPrice) + (C * CPrice) + (G *
GPrice) + (S * SPrice)
            AmountSaved = RegularPrice - DiscountPrice
            Console.WriteLine("Discount price for order = "& DiscountPrice)
            Console.WriteLine("Amount saved             = "& AmountSaved)
EndIf
```

Sample Questions

1 (a) All variables, constants and other identifiers should have meaningful names.
(i) When you performed the tasks, you used variables.
Write suitable declarations for **two** of these.
State what you used each one for.
Variable 1: .....
Use .....
.....
Variable 2: .....
Use .....
.....[4]
(ii) When you performed the tasks, you may have used constants.
Write suitable declarations for **two** of these.
State what you used each one for.
Constant 1: .....
Use: .....
.....
Constant 2: .....
Use: .....
.....[4]
(iii) It is decided to store content and weights of 100 sacks instead of single sack. Declare 2 one dimension arrays to input and store contents and weights.

Variable 1: .....
Use: .....
.....
Variable 2: .....
Use: .....
.....[4]

(b) Write an algorithm to complete **Task 1**, using **either pseudo code and** programming statements.
Pseudo code: .....
.....
.....
.....
.....
.....
.....
.....
.....
.....
Programming statements.....
.....
.....
.....
.....
.....
.....
.....
.....
..... [5]

(c) (i) Explain how you completed **Task 3**. You can include pseudocode or programming statements as part of your explanation. You should assume that Task 2 has been completed. [5]

(ii) Comment on the efficiency of your design for **Task 3**.

.....
.....
.....
.....[2]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[5]

(c) (i) Explain how you completed **Task 3**. You should assume that Task 2 has been completed. You can include pseudocode or programming statements as part of your explanation.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[6]

(ii) Comment on the efficiency of your design for **Task 3**.

.....
.....
..... [1]

(iii) Give **two sack** weights that you could use to check the validation used in **Task 1**. Explain why you chose each weight.

Weight 1 .....
Reason for choice .....
.....
Weight 2 .....
Reason for choice .....
.....[4]

2.3 Database

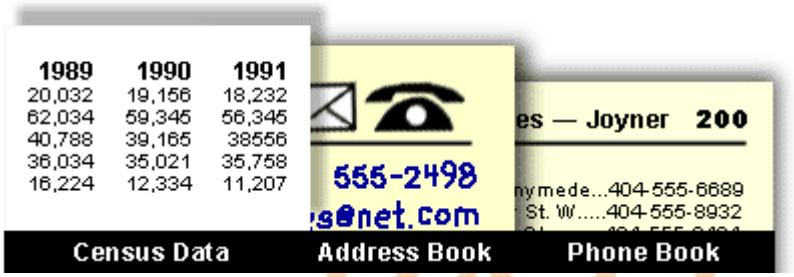
- define a single-table database from given data storage requirements
- choose and specify suitable data types
- choose a suitable primary key for a database table
- perform a query-by-example from given search criteria

A database is a collection of information organized to provide efficient retrieval. The collected information could be in any number of formats (electronic, printed, graphic, audio, statistical, combinations). There are physical (paper/print) and electronic databases.

A database could be as simple as an alphabetical arrangement of names in an address book or as complex as a database that provides information in a combination of formats.

Examples:

- phone book
- address book
- Census Bureau data



Database Management System (DBMS)

Database management system is a mechanism for manipulating data with high level command. It hides low level details such are as how data are obtained.

Database management system also has ability to search record by queries and to create reports and view data.

Entity

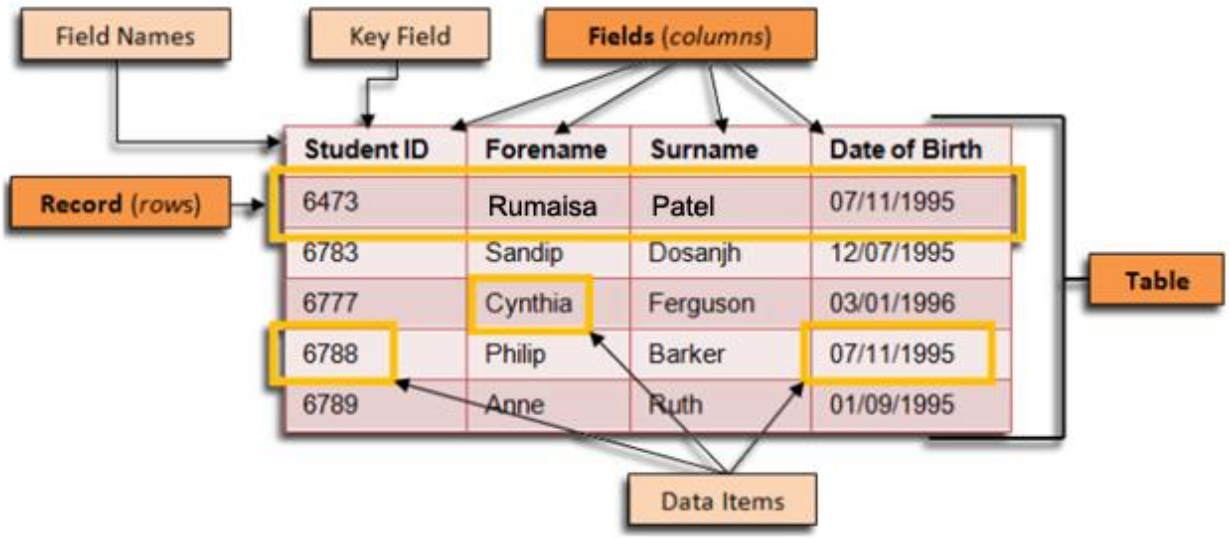
An entity is a “real world thing” about which data is held. Examples of entities include:

A customer	A product	A pupil	A supplier
A hotel room	A DVD	A flight	A holiday
A treatment	An address book	A book	A car
An order	An animal	A student	

An attribute is a feature of that entity. For example, a hotel room might have an attribute about whether it has a view or whether it is single or double. A student might have a date of birth and an address.

An entity is stored as a table in a database and an attribute becomes a field in a table. All the data about a particular entity is stored in a single table. Each data item about the

entity is a field.



Database record

Data in a database table is organised into rows (records) and columns (fields). Each record in a relational database table corresponds to an entity. In the example table of 'Students' above there are 5 records. Each record corresponds to an individual student. Note that although there are two students called Philip Barker with the same date of birth, they have different Student IDs and are different students.

Database field

An attribute is a piece of information or a characteristic of an entity. Attributes of entities are represented in database tables by fields (columns). A field stores one item of data for a record. In the table above, each student is represented in the relational database by a record and the student attributes are stored in the following fields:

- Student ID
- Forename
- Surname
- Date of Birth

Fields have the following characteristics:

- Each field in a table has a unique name. Note, however, that the same field name can occur in other tables of the same relational database.
- Each field stores a single item of data - For example, a field called Date of Birth would store no more than one date of birth value.
- Each field has a particular data type – for example, text, Boolean, integer, date/time, etc.
- Each field can have its own validation rules - these ensure that data recorded in the field is of the right type and format.

Data types

Different data types are identified so that a computer can store and process the data

appropriately.
Data types include:

- text (or string)
- number (numeric) may include:
  - Auto number
  - Currency
- date/time
- Boolean (or Yes/No).

Primary Keys

Each table has a primary key. This is a field chosen so that it can uniquely identify each record.
Sometimes an existing attribute can be used because it is unique but most of the time some sort of ID is created. Primary keys can be used to link to foreign keys in other tables. A foreign key is the primary key in a different table and it is not necessarily unique.

Example Question:

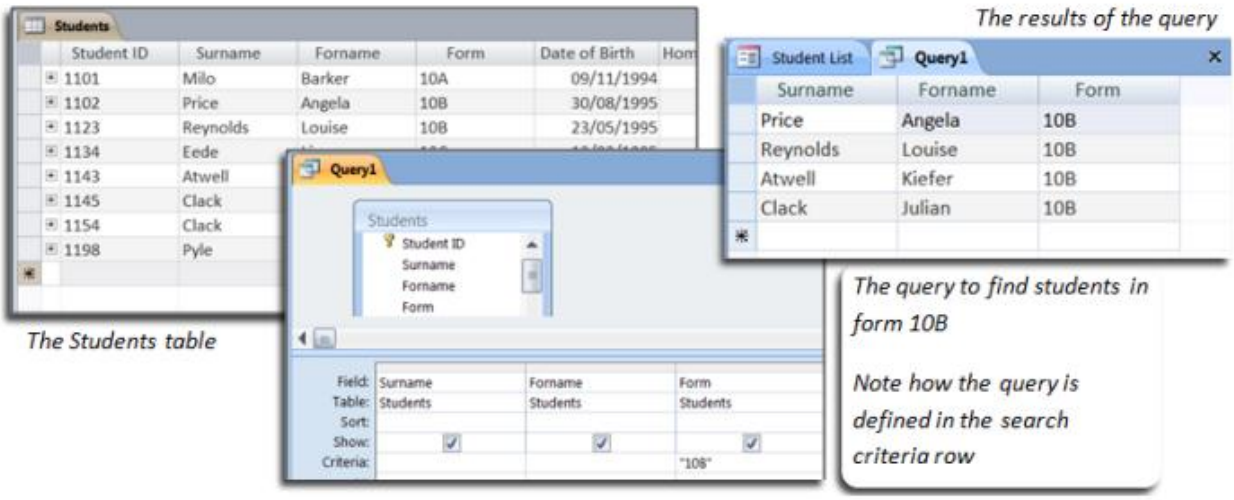
A picture gallery owner has decided to set up a database to keep information about the pictures he has for sale. The database table, PICTURE, will contain the following fields: Title; Artist; Description; Catalogue Number; Size (area in square centimeters); Price; Arrived (date picture arrived at gallery); Sold (whether picture is already sold)
(a) (i) State what data type you would choose for each field.
Title: .....
Artist: .....
Description: .....
Catalogue Number: .....
Size: .....
Price: .....
Arrived: .....
Sold: .....[4]
(ii) State which field you would choose for the primary key.....[1]

Query

The prime function of a relational database is to store data in an organised way so that users can interrogate (search) and manipulate (sort) the data. The interrogation of a database is called querying the database and a question used to interrogate the data is called a query.
Query by Example (**QBE**) is a database **query** language for relational databases. It was devised by Moshé M. Zloof at IBM Research during the mid-1970s, in parallel to the development of SQL. It is the first graphical **query** language, using visual tables where the user would enter commands, **example** elements and conditions.

Databaseuser-interface in which the user fills out a form to retrieve data. The database makes the search on the basis of the example(s) provided by the user.

The query to find students in form 10B



A complex query looks for data in two or more fields and uses the logical operators OR, AND or NOT.

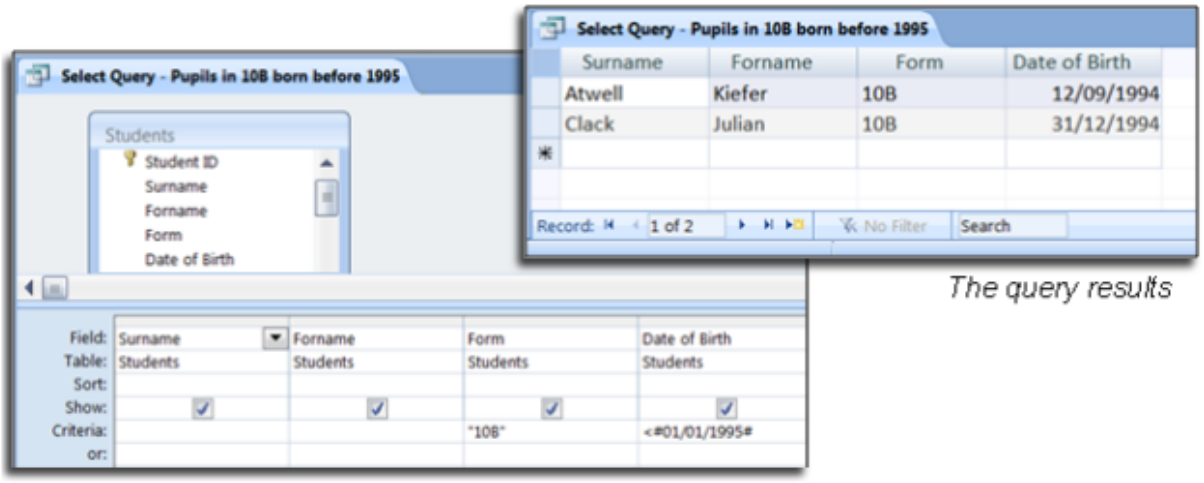
The following example uses a complex query to find all of the pupils in Form 10B who were born before 1995. This query uses the logical operator AND:

(Form = "10B") AND (Date of Birth < 01/01/1995).

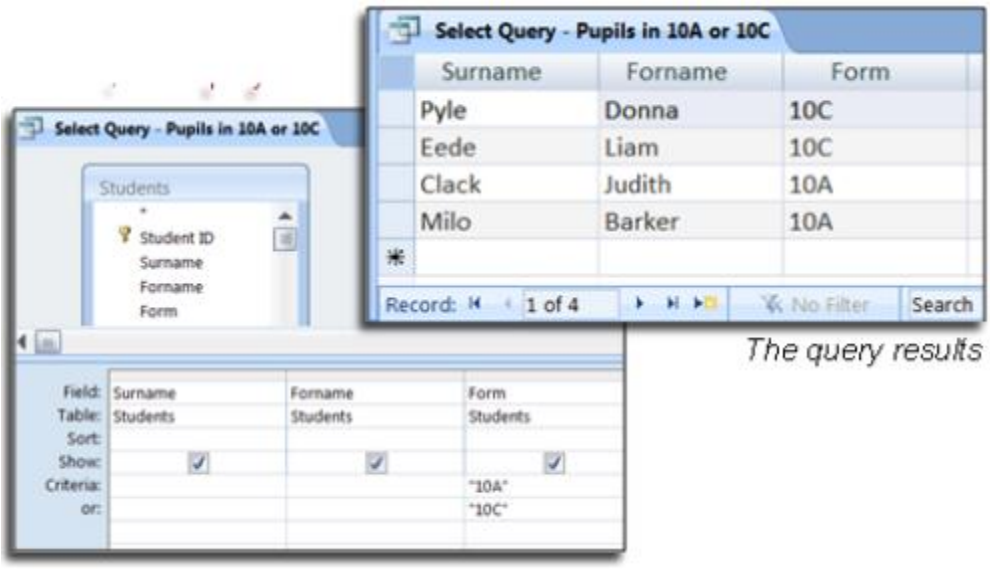
Operators can be used to refine search results.

Operator	Meaning
=	Equals
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

The query design is shown below. Note that this time there are two entries in the search criteria row Also note that this time the query has been given a meaningful name ***("Select Query – Pupils in 10B born before 1995").*** This saves other database users from unnecessarily creating the same query.



Below is a new complex query that uses the logical operator OR to find pupils who are in Form 10A or Form 10C: (Form = “10A” OR “Form = “10C”) This time, in the query definition there will be two criteria lines. The query and its results are shown below:



The query results

Wildcards in Queries

Wildcard characters can be used in database queries. For example you may want a list of all pupils born in November, or all of the pupils whose surname starts with a ‘C’. Wildcard searches allow you to specify the part of the data that you know and leave the data handling software to fill in the blanks.

Surname Like “C*” would find all records where the surname begins with a C**.**

Example Question:

A database was set up to compare oil companies. A section of the database is shown below:

Code	Name of company	No of employees	No of countries	Head office	Profits (billion $)	Share price ($)
AR	Arrows	60 000	30	Americas	8.0	39.00
GZ	Gazjeti	35 000	4	Asia	5.0	44.50
KO	Konoco	40 000	22	Americas	10.0	18.55
OS	Oilbras	56 000	11	Americas	4.0	59.60
SD	Sand Oil	102 000	51	Europe	12.0	15.30
SN	Southern Oil	50 000	15	Americas	11.0	10.90
ST	Static Oil	80 000	31	Americas	10.0	52.05
SU	Summation	70 000	40	Europe	9.0	30.40
WP	Wasp Petrol	90 000	44	Europe	15.0	92.80

(a) Complete the query-by-example grid below to show no. of employees, head office and profit, have a No of countries greater than 40 and Head office is Europe?

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

(b) Output of the above query will be: .....
..... [2]

(a) Complete the query-by-example grid below to show name of oil companies and code have a share priceless than $50 or whose profits were greater than 8 billion dollars?

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

(b) Output of the above query will be: .....
..... [2]

A survey of motorways was carried out and a database was produced. A section of the database is shown below.

Motorway ID	Length (km)	Cars per day	Toll charge per km ($)	Number of lanes
M1	100	50 000	0.60	2
M2	210	75 000	0.40	3
M3	180	60 000	0.50	4
M4	40	20 000	0.30	3
M5	25	15 000	0.10	2
M6	100	40 000	0.70	4
M7	30	10 000	0.40	2
M8	150	60 000	0.60	4

(a) How many fields and how many records are shown?

(i) number of fields

.....

(ii) number of records

..... [2]

(b) Complete the query-by-example grid below to show Motorway ID and Lanes if the following search condition was used?

(Length (km) > 100) AND (Number of lanes > 3)

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

(c) Output of the above query will be:.....[1]

(d)Complete the query-by-example grid below to show the motorways where the number of cars perday exceeds 50 000 or the toll charge per kilometer is greater than $0.50?

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

(e) Output of the above query will be: .....[2]

Methods of validating data as it is input

A validation check is a rule that is built into a database to check that the data entered is:

- Sensible
- Reasonable
- Within acceptable boundaries
- Complete

It does NOT mean that the data is actually correct, that requires verification.

There are a number of different validation rules that can be used in a database:

Type Checks - Field data types provide a basic method of validation. Field data types are assigned to fields during the creation of the database table and data types such as Numeric, Boolean, Date/Time and Image restrict what can be entered. If a user tries to enter text in a date field or alphabetic characters in a numeric field, their entry will be rejected.

Range checks - these are used to limit the range of data a user can enter. The 'day' part of a date must be in the range 1 to 31. An exam grade should be in the range 'A'...'G' or

'U'.

Check digits - this type of check is used with numbers. An extra 'check digit' is calculated from the numbers to be entered and added to the end. The numbers can then be checked at any stage by recalculating the check digit from the other numbers and seeing if it matches the one entered. One example where a check digit is used is in the 10 digit ISBN number which uniquely identifies books.

The last number of the ISBN is actually the check digit for the other numbers, for example - the ISBN 0192761501.

Presence checks - these simply check that an entry has been made in a particular field i.e. a null value (empty field) is not permitted. Usually, not every field in a record needs to be filled in, however there are likely to be some that must have a value and the presence check means that the system will not allow the record to be saved until an entry is made. An application for a passport must have the applicant's surname.

Length Checks - All alphanumeric data has a length. A single character has a length of 1 and a string of text such as “Hello World” has a length of 11 (spaces are counted in text strings). A length check ensures that such data is either an exact length or does not exceed a specified number of characters. Mobile phone numbers are stored as text and should be 11 characters in length.

Lookup - A lookup check takes the value entered and compares it against a list of values in a separate table. It can then return confirmation of the value entered or a second list based on the value. One use of lookups restricts users to pre-defined input using drop-down lists. A user is forced to use a list box to select from a predetermined list of valid values.

Input Masks - Certain alphanumeric fields in a database may require entry in a particular format, e.g. a mixture of numbers and letters. Simple examples of this are dates or text of a specific length. More complex examples include data items such as postcodes, National Insurance numbers, driving license numbers or product codes. Most databases allow formats to be defined for database fields by an input mask, which defines the valid characters permitted in a field.

Common input mask codes	
N	a digit between 0 and 9 must be entered
#	an entry is optional, but it must be a digit between 0 and 9
L	a letter between a to z must be entered
?	an entry is optional, but it must be a letter from a to z
A	a letter or digit must be entered
a	an entry is optional, but it must be a letter or digit

Example Question:

Q 1) A hospital holds records of its patients in a database. Four of the fields are:

- date of visit (dd/mm/yyyy)
- patient's height (m)
- 8-digit patient ID
- contact telephone number

The presence check is one possible type of validation check on the data. For each field, give another validation check that can be performed. Give an example of data which would fail your named validation check.

A different validation check needs to be given for each field.

Field Name	Name of validation check	Example of data which would fail the validation check
Date of visit		
Patient's height		
Patient ID		
Contact telephone number		

Marking scheme

Field Name	*Name of validation check*	*Example of data which would fail the validation check*
Date of visit	*format check* *type/character check*	*e.g. 2012/12/04* *e.g. 3rd March 2012*
Patient's height	*range check* *limit check*	*can't be < 0 or > 2.5m* *e.g. –5, five* *e.g. 8, -3,*
Patient ID	*type check* *length check* *range check*	*(can't be < 0 or > 99999999)* *e.g. 3142ABCD* *e.g. 2131451, 136498207* *e.g. –3, 851341625*
Contact telephone number	*length check* *type/character check* *format check*	*e.g. 0773141621834* *e.g. 7H215GD* *e.g. 01223/123456/8901234*

Q2) The terms file, record and field are used in databases. Explain the meaning of each term and explain the connections between them; you may wish to include a diagram.

.....

.....

.....

Q3) An estate agent uses a computer system to store details about properties for sale. A section of the properties file is shown below:

Property	Type	Area	Price
4217	D	Lake	99.92
4219	D	Park	200.00
4220	S	Lake	105.50
4221	F	Park	175.25
4222	F	Town	75.00

a Why is coded data used for Type?

.....

.....

b What is the data type for Price?

.....

.....

c What is meant by a key field? Which is the key field?

.....

.....

Q4) Explain the following terms:

database: .....

.....

file: .....

.....

record .....

.....

Q5) A shop sells bicycles. Information about each bicycle in stock is held in a computer file. Part of the information is shown in the table below.

Make	Model	Type	Colour	StockNo	Price
Smith	Sprint	Racing	B	1471	$200.00
Thompson	Maxi	Racing	B	1329	$185.00
Brown	Speed	Racing	G	1654	$200.00
Davies	Swift	Mountain	R	1972	$0.15
Smith	Swallow	Racing	Y	2015	$300.00
Thompson	Fly	Mountain	B	2149	$400.50
Brown	Panther	Racing	Y	3249	$670.00

How many fields are shown?

.....

.....

How many records are shown?

.....

.....

Give a reason why the shop would need to delete a record from the bicycle stock file.

.....

.....

Give a reason why the shop would need to add a record to the bicycle stock file.

.....
.....

Explain how the shop can print a list of mountain bikes.

.....
.....

There is an error in the table. Which item is wrong? How could this error have been avoided?

.....
.....

Which is the key field?

.....

Q6) A video club hires films to members. The club uses separate database f les to store details about films, members and the films hired.

a Name four suitable items of information for the FilmsHired database file, specify the field type and give an example of the data.

.....
.....

b Name four suitable items of information for the Members database file, specify the field type and give an example of data.

.....
.....

Q7) A teacher uses a database to store the marks of pupils from all year 9 classes.

(a) PUPIL and CLASS are two entities used in this database.

Explain the term entity.

.....
.....
.....
..... **[2]**

(b) The data for the first four pupils in the PUPIL table is shown below.

PupilNumber	Surname	FirstName	ClassCode
A01	Adams	Michelle	9DK
A02	Ali	Mohammed	9BH
A03	Ali	Shirelle	9DK
A04	Azor	Michelle	9FT

(i) State the primary key for the PUPIL table and explain your answer.

Primary Key .....**[1]**

Explanation

.....
..... **[2]**

(ii) The database also contains a CLASS table. The primary key for the CLASS table is ClassCode.Explain why ClassCode has also been included in the PUPIL table.

.....
.....
..... **[3]**

Q8) An MP3 player contains a database of songs. This database has only one table. A sample of the data in this table is shown below.

TrackNo	Artist	Song	Length	TimesPlayed	Protected
001	Dave Eade	Holidays	3.7	3	True
002	Tail	Seeing You	2.7	0	True
003	Dave Eade	Truly Cool	4	11	False
004	Aries	Love	1.9	0	True
005	Mc Nail	Skit	0.4	0	False
006	The Flies	Skit	0.6	4	False
007	Mc Nail	Game Over	2.7	1	True

(c) State the most appropriate data type for each of the fields shown below:

Field	Data Type
Song	
Length	
TimesPlayed	
Protected	

(b) i. State how many records are there? _____
ii. State how many fields are there in each record? _____
(c) The mp3 player allows users to create playlists by using queries.
For example if the query used is
Artist =”Dave Eade”
The mp3 player will play tracks number 001 and 003.
(i) State the TrackNo of the songs that will be played using each of the following queries.
Length>2
.....
.....
(Artist=”Mc Nail”) OR (Protected=False)
.....
.....
(Song=”Skit”) AND (TimesPlayed>0)
.....
(ii) Write down the queries that will be select all songs over 2.5 minutes, which have never been played.
.....

Examination Questions:

Q1) Winter 2009

A radio station keeps a database of all its music CDs. Here is part of this database:

Reference Number	CD title	number of tracks	special edition	CD length (mins)	number of hit tracks
1111	Afternoon Glory	12	N	55	1
1112	Stone Tulips	10	N	42	3
1113	Aftermath	8	N	33	0
1114	Major Peppers	15	Y	72	5
1115	Seaside	9	N	40	2
1116	Lookout	12	N	62	2
1117	Future Dreams	11	N	60	3
1118	Moonlight	14	Y	70	2

(a) How many records are there in the database section?

..... [1]

(b) If the following query was input:

(CD length (mins)<60) AND (number of hit tracks>1)

Using Reference Number only, write down which data items would be output.

..... [1]

(c) Write down a query to select which CDs are special edition or have more than 10 tracks.

..... [1]

(d) The database is sorted in descending order on CD length (mins). Using Reference Number only, write down the order of the records following this sort.

..... [1]

(e) The radio station has a phone-in service where a listener texts the title of the CD on their mobile phone. The popularity of each CD is then known and which CDs the radio station should play.

(i) How would this information be stored?

..... [1]

(ii) How could this information be linked to the database?

..... [1]

Q2) Winter 2010:

A database has been set up to store information about aircraft. A section is shown below.

Ref No	Aircraft Name	Max Weight (kg)	Length (m)	Wing Span (m)	Max Speed (kph)
1001	An-225 Cossack	600 000	84	88	850
2001	Airbus A380F	591 950	73	80	951
3001	C-5 Galaxy	381 000	76	68	845
3002	Boeing 777-600	351 500	74	65	930
2002	Airbus A340-600	366 000	75	63	877
3003	Boeing 747	397 000	71	64	967
3004	Boeing 777	660 000	74	61	893
2003	Airbus A330-300	234 000	63	60	800
3005	Boeing 767	204 100	61	52	914
3006	B52 Fortress	221 400	49	56	927
3007	Boeing 757	123 400	54	38	914

(a) How many fields are in each record?

(b) Using Ref No only, what records would be output if the following search condition was entered:

(Max Weight(kg) > 350 000) AND (Wing Span(m) < 66)?

(c) Write down the search condition to find out which aircraft have a length greater than 74 metres or have a maximum speed less than 900 kph.

14 An international bank keeps records of customer account details on a computer.

(a) It is necessary on occasions to:

- delete records
- amend records
- insert records

Give one example of when each of the above would need to be done. [3]

(b) A section of one record is shown below:

Frederick Parez	Rua Silva Paulet	5151 315 000	34	20 – 15 – 00	Br
name	address	telephone number	age	branch	country

(i) The branch and country are coded. Give a reason for this. [1]

(ii) One of the six fields is not appropriate.

Name this field and give a reason for your choice. Suggest an improved field.

Name of field

Reason for choice

Improved field choice [3]

Q3) Summer 2011:

A database showing the population of world cities has been produced. A section of the database is shown below.

Ref No	Name of City	Country	Area	City Population (m)	Urban Population (m)	Capital
1	Tokyo	Japan	Asia	33.2	34.1	Yes
2	New York	USA	America	17.8	21.9	No
3	Sao Paulo	Brazil	America	17.7	20.2	No
4	Seoul	S Korea	Asia	17.5	22.3	Yes
5	Mexico City	Mexico	America	17.4	22.7	Yes
6	Osaka	Japan	Asia	16.4	16.8	No
7	Manila	Philippines	Asia	14.8	14.9	Yes
8	Mumbai	India	Asia	14.4	19.7	No
9	Jakarta	Indonesia	Asia	14.3	17.2	Yes
10	Calcutta	India	Asia	12.7	15.6	No

(a) How many records are shown above?

(b) Using Ref No only, which records would be found if the following search condition was typed in

(Country = “India” OR Area = “America”) AND (Capital = “No”)

(c) Write a search condition to find the cities in Asia with a city population greater than 17 million OR an urban population greater than 20 million.

(d) Give one advantage of using Y or N rather than Yes or No in the Capital column.

Q4) Winter 2011:

An airport has a number of hotels nearby. A database has been set up to give customers information to allow them to select a hotel.

Hotel Ref	Name of hotel	No. of stars	No. of rooms	Hotel parking	Price per person ($)	Distance from airport (km)
H41	The Grand	3	45	Y	65	11
K22	Sleepy Inn	2	15	N	45	10
N15	Britannia	5	140	Y	150	4
L44	Beach Hotel	4	62	N	85	8
H30	Sea View	3	38	N	60	4
H21	Pyramid	3	25	N	70	5
N21	Superior	5	120	Y	200	2
K14	Travellers	2	15	N	45	10

(a) How many records are shown in the database?

(b) Which field in each record must be unique?

(c) The following search condition was typed in:

(No. of stars > 3) OR (Hotel parking = Y)

Using Hotel Ref only, which records would be found?

(d) Write down the search condition to find which hotels were less than 10 km from the airport and charged under $100 per person.

(e) The database was sorted into descending order using No. of rooms.

Using Hotel Ref only, write down the sorted order of records.

Q5) Winter 2011:

A database has been set up to show details about countries. Part of the database is shown below.

Country code	Country	Continent	Area (millions sq km)	Population (millions)	Coastline	Currency
CH	China	Asia	9.6	1320	Yes	yuan
IN	India	Asia	3.8	1150	Yes	rupee
PO	Poland	Europe	0.3	39	Yes	zloty
BO	Bolivia	America	1.1	9	No	boliviano
TI	Tibet	Asia	1.2	2	No	yuan
BR	Brazil	America	8.5	192	Yes	real
RO	Romania	Europe	0.2	22	No	leu
SA	Saudi Arabia	Asia	2.2	28	Yes	riyal
ZA	Zambia	Africa	0.7	12	No	kwacha

(a) How many fields are in each record?

(b) Using Country code only, what would be output if the following search condition was used?

(Population (millions) > 1000) OR (Continent = “Asia”)

(c) Write down a search condition to find which countries have a land area less than 3 million square km and also have a coastline.

(d) If the database was sorted in descending order of population size, using Country code only, what would be the order of countries in the database?

Q6) Summer 2012:

A database was set up to show the properties of certain chemical elements. Part of thedatabase is shown below.

Name of element	Element Symbol	Atomic Number	Atomic Weight	Melting Point (C)	Boiling Point (C)	State at room temp
oxygen	O	8	16	- 218	- 183	gas
iron	Fe	26	56	1538	2861	solid
mercury	Hg	80	201	- 38	356	liquid
bromine	Br	35	80	- 7	59	liquid
osmium	Os	76	190	3033	5012	solid
caesium	Cs	55	133	28	671	solid
gallium	Ga	31	70	30	2204	solid
argon	Ar	18	40	- 189	- 186	gas
silver	Ag	47	108	961	2162	solid

(a) How many fields are in each record?

(b) The following search condition was entered:

(Melting Point (C) < 40) AND (Atomic Weight > 100)

Using Element Symbol only, which records would be output?

(c) We need to know which elements have an atomic number greater than 50 and are solid at room temperature.

Write down the search condition to find out these elements.

(d) The data are to be sorted in descending order of Boiling Point (C).
Write down the new order of records using the Element Symbol only.

Q7) Winter 2012:

A database was set up showing the largest ocean-going liners. Part of the database isshown below.

Liner ID	Year built	Gross Tonnage	Country of Registration	Country of Construction
OA	2009	225282	Norway	Finland
IN	2008	154407	Norway	Finland
QM	2004	148528	UK	France
EX	2000	137308	Norway	Finland
VO	1999	137276	Norway	Finland
GP	1997	108865	UK	Italy
DE	1996	101509	USA	Italy
SP	1995	77499	UK	Italy
SO	1988	73192	Norway	France
FR	1972	66343	France	France
QE	1940	86673	UK	UK
NO	1935	79280	France	France
MJ	1922	56561	UK	Germany
TI	1912	46329	UK	UK
MA	1907	31938	UK	UK

(a) How many records are shown in the above part?
(b) Using Liner ID only, what would be output if the following search condition was typed in:
(Year built < 2000) AND (Country of Registration = Country of Construction)?
(c) Write the search condition to find out which liners have a gross tonnage larger than 80 000 or are registered in the UK.

Q8) Winter 2013 P12:

A motor car manufacturer offers various combinations of

- seatcolours
- seat materials
- car paint colours

A database was set up to help customers choose which seat and paint combinations were possible.

	seat material			car paint colours						
code	cloth	leather	seat colour	*white*	*red*	*black*	*blue*	*green*	*silver*	*grey*
CB	Y	N	black	Y	Y	Y	Y	Y	Y	Y
LB	N	Y	black	N	Y	N	N	N	Y	Y
CC	Y	N	cream	N	Y	Y	Y	N	N	N
LC	N	Y	cream	N	Y	Y	Y	N	N	Y
CG	Y	N	grey	N	Y	Y	Y	Y	Y	N
LG	N	Y	grey	N	Y	N	Y	N	Y	Y
CR	Y	N	red	Y	N	Y	N	N	Y	Y
LR	N	Y	red	Y	N	Y	N	N	Y	Y
CL	Y	N	lime	N	N	N	Y	N	N	N
LL	N	Y	lime	N	N	Y	Y	Y	N	N

(NOTE: N = no, not a possible combination, Y = yes, combination is possible)

(a) How many records are shown in the database? [1]

(b) The following search condition was entered:

(cloth= "Y") AND (blue = "Y")

Using code only, which records will be found? [2]

(c) A customer wanted to know the possible combinations for a car with leather seats and either silver or grey paint colour.

What search condition would need to be input? [2]

(d) A customer decided to buy a green car. He wanted to know which seat colours and seat materials were not a possible combination with green paint.

What search condition would he need to enter? [1]

(e) Give one advantage of using the codes Y and N in the database rather than using Yes and No. [1]

Q9)Winter 2013 P13:

A database was set up to keep track of goods in a shop. A section of the database is shown below.

Item code	Number in stock	Re-order level	Price of item ($)	Value of stock ($)	Items ordered
1113	155	200	1.50	232.50	Yes
1124	84	50	2.50	210.00	No
1200	30	60	5.00	150.00	Yes
1422	600	500	1.00	600.00	No
1515	90	100	2.00	180.00	No
1668	58	50	4.00	232.00	No
1801	60	100	8.00	480.00	No
1844	195	200	1.50	292.50	Yes

(a) How many records are shown in this section of database? [1]
(b) (i) Using Item code only, what would be output if the following search was carriedout:
(Number in stock <Re-order level) AND (Items ordered = "No") [2]
(ii) What useful information does this search produce? [1]
(c) Write a search condition to locate items costing more than $2.00 or have a stock value exceeding $300.00. [2]

Q10) Summer 2014 P11:
A hospital holds records of its patients in a database. Four of the fields are:
• date of visit (dd/mm/yyyy)
• patient's height (m)
• 8-digit patient ID
• contact telephone number
The presence check is one possible type of validation check on the data. For each field, give another validation check that can be performed. Give an example of data which would fail your named validation check.
A different validation check needs to be given for each field.

Field Name	Name of validation check	Example of data which would fail the validation check
Date of visit		
Patient's height		
Patient ID		
Contact telephone number		

Introduction to Logo

LOGO is a programming language that was developed at the MIT Artificial Intelligence Lab in the 1970s. It was designed to be used as an introduction for people toboth programming and artificial intelligence. However, while it is easy to learn,Logo is a powerful language. Once you know the basics, Logo can be used to doextremely complicated things. Logo code has been used in telecommunications, multimedia software and robotics. Finally Logo is FUN!

Turtle?

When Logo started, the Turtle was a robotic creature which sat on the floor and wasdirected to move around by a user typing commands on a computer. Today the turtleis represented by an icon on the screen and can be made to draw on the screen usingthe same commands. In some environments the turtle looks like a turtle (with head,tail and feet) in others represent the turtle with a triangle.

Logo Commands

The following are some of the most use full commands in theLogo Language, which you will want to become familiar with:

FORWARD - Follow this command with a number (such as: 10 or 1000.) A smallnumber will cause the turtle to move forward a short distance. A largernumber will cause it to move further. If you select a large enough number the turtle will go off the canvas and wrap around to the other side.

BACK- Follow this command with a number, the same as FORWARD, only thistime the turtle will move backwards.

RIGHT - Follow this command with a number between 0 and 360. The turtle willturn right specified number of degrees.

LEFT - Follow this command with a number between 0 and 360. This command isthe same as RIGHT only it will turn the turtle left, not right.

PENUP - This command will cause the turtle to pick up its “pen” up so that youcan move the turtle without drawing a line.

PENDOWN - This is the command you would use to put the “pen” back down soyou can draw again.

SETPENCOLOR - You can change the colour your turtle draws in. Follow the command with a number to get different colours. For example
“SETPENCOLOR 0” would give you a black pen.

CLEAN- This command will erase the canvas

HOME- This command will move the turtle back to the centre of the canvas

The Repeat Command

You can get your turtle to do one (or several) things repeatedly, without typing them again and again using the REPEAT command. Typing

```
REPEAT 4 [FORWARD 10]
```

Would cause the turtle to move forward 10 spaces, 4 time. So, in total the turtlewould move forward 40 spaces.Now Try These. Type:

```
REPEAT 4 [FORWARD 50 RIGHT 90]
```

You should get a square. For a bigger square try replacing 50 with 100 or 200.Type:

```
REPEAT 360 [FORWARD 2 LEFT 1]
```

You should get a circle. For a smaller circle try replacing 2 with 1. Can you make abigger one?

Your First Program

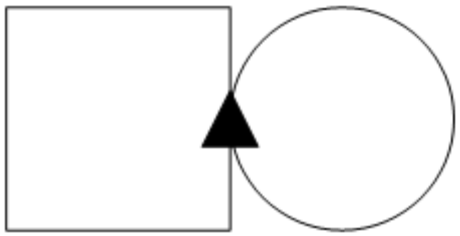
That's a lot to type every time you want to make a square or circle though. Can it beeasier? YES. You can teach Logo what a square (or a circle, or a flower) is by making it program.Try typing:

```
TO SQUARE
REPEAT 360 [FORWARD 80 LEFT 90]
END
```

Now type SQUARE and see what happens.

How would you write the program CIRCLE?

How would you write the program CIRCLE_AND_SQUARE to make a drawingthat looks like this (where the black triangle is the turtle at the end)?



```
TO CIRCLE_AND_SQUARE
HOME
CLEAN
CIRCLE
FORWARD 52
REPEAT 3 [ LEFT 90 FORWARD 104]
LEFT 90
FORWARD 52
END
```

CIRCLE was defined as:

```
TO CIRCLE
REPEAT 360 [FORWARD 1 RIGHT 1]
END
```

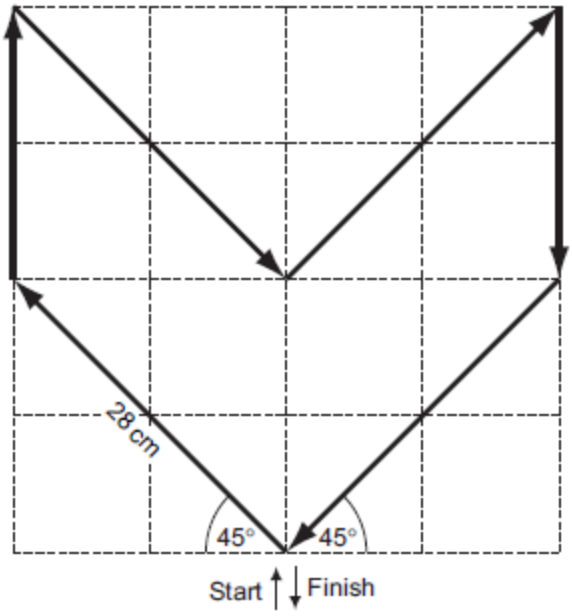
Summer 2014 P11

A floor turtle can use the following instructions.

Instruction	Meaning
FORWARD *x*	Move *x* cm forwards
BACKWARD *x*	Move *x* cm backwards
LEFT *t*	Turn left *t* degrees
RIGHT *t*	Turn right *t* degrees
REPEAT *n*	Repeat next set of instructions *n* times
ENDREPEAT	Finish repeated instructions
PENUP	Lift the pen
PENDOWN	Lower the pen

Each square is 10 cm by 10 cm

Each diagonal line is 28 cm long



Complete the set of instructions to draw the above shape.

Pen Down

…….

Left 45

…….

……

……

……

……

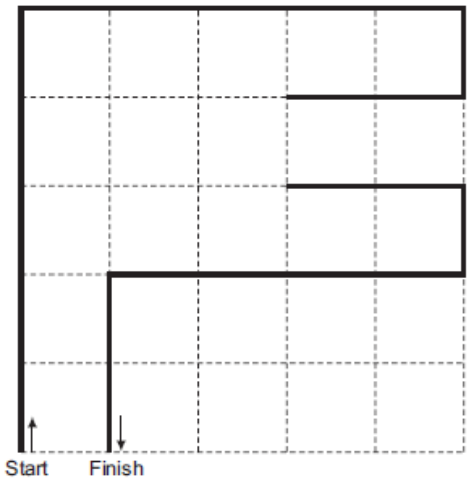
……

Summer 2014

A floor turtle uses the following commands:

command	description
FORWARD *n*	Move *n* cm in a forward direction
BACKWARD *n*	Move *n* cm in a backward (reverse) direction
RIGHT *t*	Turn right through *t* degrees
LEFT *t*	Turn left through *t* degrees
PENUP	Lift the drawing pen up
PENDOWN	Lower the drawing pen
REPEAT *x*	Repeat the next set of instructions *x* times
ENDREPEAT	Finish the REPEAT loop

In the following grid, each of the squares measures 10 cm by 10 cm:



Complete the set of instructions to draw the shape shown above (in bold lines).

Pen Down .....

Repeat 2.....

.....

.....

.....

.....

.....

.....

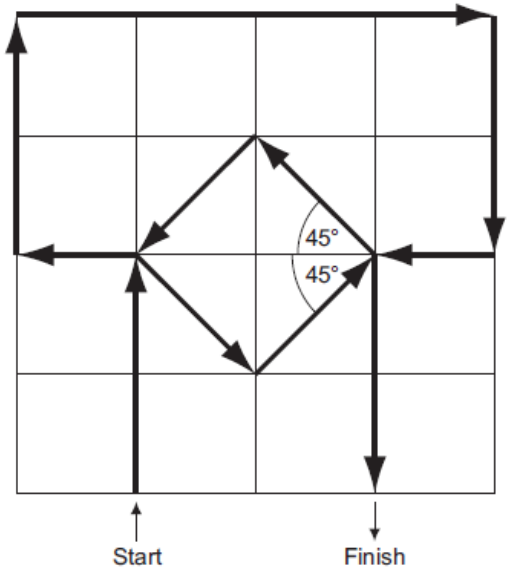
Summer 2012

A floor turtle can use the following instructions.

Instruction	Meaning
FORWARD *x*	Move *x* cm forwards
LEFT *t*	Turn left *t* degrees
RIGHT *t*	Turn right *t* degrees
REPEAT *n*	Repeat next set of instructions *n* times
ENDREPEAT	Finish repeated instructions
PENUP	Lift the pen
PENDOWN	Lower the pen

Each square = 10 cm by 10 cm

Each diagonal line = 14 cm



Complete the set of instructions to draw the above shape.

Pen Down .....

Forward

20.....

Left 90 .....

.....

.....

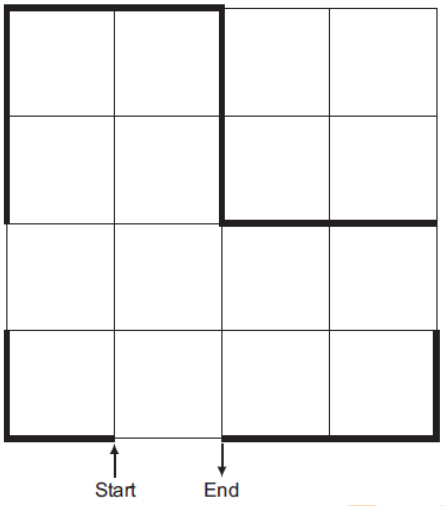
.....

Summer 2010

A floor turtle can use the following instructions:

Instruction	Meaning
FORWARD ***d***	Move ***d*** cm forward
BACKWARD ***d***	Move ***d*** cm backward
LEFT ***t***	Turn left ***t*** degrees
RIGHT ***t***	Turn right ***t*** degrees
REPEAT ***n***	Repeat the next set of instructions ***n*** times
ENDREPEAT	End of REPEAT loop
PENUP	Raise the pen
PENDOWN	Lower the pen

(In the following grid, each square is 10 cm by 10 cm.)



Complete the set of instructions to draw the above shape.

Pen Down .....

Left 90 .....

Forward

10.....

Right 90 .....

.....

.....

.....

.....

Winter 2014 P12

Six statements and six values are shown below.
Each statement will generate one possible value.
Draw a line to link each statement to its correct value.

statement	value
number of times the following loop operates: count = 1 **repeat** **input** x count = count + 1 **until** count = 5	1
the number of bits that make up a byte	4
base 10 (denary) value of the following binary number: 32 16 8 4 2 1 0 0 1 1 1 1	5
the number of tracks on the single side of a CD-R	8
number of minutes to upload a 75 Mbyte file at 2 megabits/ second upload speed	10
If there are 2^X bytes in a Kbyte, what is the value of **X**?	15

Winter 2014 P13

Six statements and six values are shown below.
Each statement will generate one possible value.
Draw a line to link each statement to its correct value.

statement	value
number of possible binary input combinations for a 2-input logic gate circuit	0
output from the logic gate: 	1
base 10 (denary) value of the following binary number: 4 2 1 1 1 0	4
what is the output from the algorithm: y = 1 **for** x = 1 **to** 4 y = y * x **next** x **print** y	6
number of bytes formed from 8 bits	20
If there are 2^x bytes in a Mbyte, what is the value of **X**?	24

Past Papers

Summer 2015 P21& 23

Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1

A data logger records the temperature on the roof of a school twice a day, at midday and midnight. Input and store the temperatures recorded for a month. You must store the temperatures in two one dimensional arrays, one for the midday temperatures and one for the midnight temperatures. All the temperatures must be validated on entry and any invalid temperatures rejected. You must decide your own validation rules. You may assume that there are 30 days in a month.

TASK 2

Calculate the average temperature for midday and the average temperature for midnight. Output these averages with a suitable message for each one.

TASK 3

Select the day with the highest midday temperature and the day with the lowest midnight temperature. Then output each of these temperatures, the corresponding day and a suitable message. Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

1 (a) All variables, constants and other identifiers should have meaningful names.

(i) In **Task 1**, you had to store the midday temperatures and midnight temperatures inarrays.

Write suitable declarations for these **two** arrays.

.....

.....[2]

(ii) It has been decided to record the temperatures for one week rather than one month.

Write the new array declarations that you would use.

.....

.....[1]

(iii) Declare **two** other variables that you have used and state what you used each one for.

Variable 1 .....

Use .....

Variable 2 .....

Use .....[4]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements**or**a flowchart. Use temperatures for **one week** only. You should assume that Task 1 hasalready been completed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(c) **(c)** Give a set of midday temperature data, for a week that could be used to check your validationrules for **Task 1**. Explain why you chose this data set.

Data set: .....

Reason for choice: .....

..... [2]

(d) Explain how you select the day with the highest midday temperature (part of **Task 3**). You mayinclude pseudocode or programming statements as part of your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[6]

Section B

2 Read this section of program code that should input 10 positive numbers and then output the smallest number input.

```
1 Small = 0
2 Counter = 0
3 REPEAT
4 INPUT Num
5 IF Num< Small THEN Num = Small
6 Counter = Counter + 1
7 PRINT Small
8 UNTIL Counter < 10
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 .....

.....

2 .....

.....

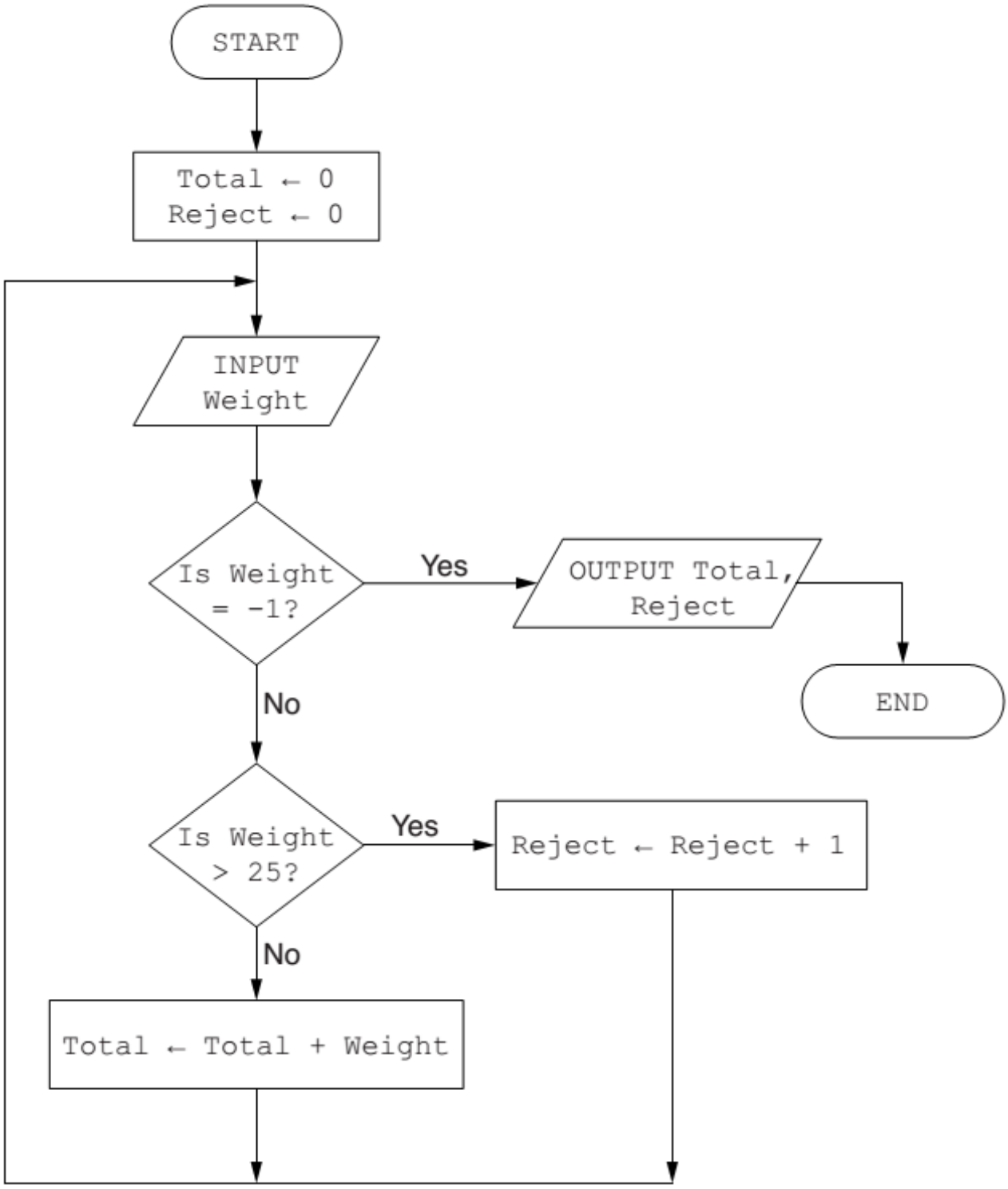
3 .....

.....

4 .....

..... [4]

3 The flowchart below inputs the weight of a number of parcels in kilograms. Parcels weighing morethan 25 kilograms are rejected. A value of –1 stops the input.
The following information is output: the total weight of the parcels accepted and number of parcelsrejected.



Complete the trace table for the input data:

1.8, 26.0, 7.0, 11.3, 10.0, 2.5, 25.2, 5.0, 19.8, 29.3, –1 [5]

Total	Reject	Weight	OUTPUT

4 Five data types and five data samples are shown below.

Draw a line to link each data type to the correct data sample. [4]

Data type	Data sample
Integer	'a'
Real	2
Char	2.0
String	True
Boolean	"Twelve"

5 Explain the difference between a variable and a constant in a program.

.....

.....

.....

.....[2]

6 Identify **three** different loop structures that you can use when writing pseudocode.

1.....

2 .....

3..... [3]

7 A database, PROPERTY, was set up to show the prices of properties for sale and the features of each property. Part of the database is shown below.

Property Type	Brochure No	Number of Bedrooms	Number of Bathrooms	Garden	Garage	Price in $
Bungalow	B17	7	4	Yes	Yes	750,000
Apartment	A09	2	1	No	No	100,000
House	H10	4	2	Yes	No	450,000
House	H13	3	2	Yes	No	399000
Apartment	A01	2	2	No	Yes	95000
Apartment	A16	1	1	No	No	150000
House	H23	3	1	No	Yes	250000
House	H46	2	1	Yes	Yes	175000

(a) Give the number of fields that are in each record.

.....[1]

(b) State which field you would choose for the primary key.

.....

Give a reason for choosing this field.

.....

.....[2]

(c) State the data type you would choose for each of the following fields.

Garage .....

Number of Bedrooms .....

Price in $ .....[3]

(d) The query-by-example grid below selects all houses with more than 1 bathroom and more than 2 bedrooms.

Field:	Property Type	Number of Bedrooms	Number of Bathrooms	Price in $	Brochure No
Table:	PROPERTY	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Sort:				Ascending	
Show:	☐	☐	☐	☑	☑
Criteria:	= 'House'	>2	>1		
or:					

Show what would be output.

.....

.....[2]

(e) Complete the query-by-example grid below to select and show the brochure number, property type and price of all properties with a garage below $200,000.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

[4]

Marking Scheme

Section A

1 (a) (i) Many correct answers, they must be meaningful. These are examples only.

– MiddayTemperature[1:30]
or MiddayTemperature[0:29]
or MiddayTemperature[30]
or MiddayTemperature[29]
or MiddayTemperature[] (1 mark)
– MidnightTemperature[1:30]
or MidnightTemperature[0:29]
or MidnightTemperature[30]
or MidnightTemperature[29]
or MidnightTemperature[] (1 mark) [2]

(ii) Answers, must match above and the upper bound should have been changed from30 to 7 or 29 to 6 or no change if not used. These are examples only.

– MiddayTemperature[1:7] MidnightTemperature[1:7]
or MiddayTemperature[7] MidnightTemperature[7] [1]

(iii) Any two variables with matching reasons, 1 mark for the variable and 1 markfor the matching reason. The variables and the matching reasons must relate tothe tasks in the pre-release. There are many possible correct answers these areexamples only.

Variable – Counter: (Integer)
Reason – to use as a loop counter when entering the temperature
Variable – HighNoon: (Real)
Reason – to store the highest midday temperature [4]

(b) If loop used
– initialisation before loop
– loop
– running total inside loop
– calculation of average outside loop
– output of average with message outside loop (Max 4 marks)
– completion of at least 3 of initialisation, running total, calculation of average and output of average with message for both midday and midnight (1 mark) [5]

sample algorithm:

```
MiddayTotal ← 0; MidnightTotal ← 0
FOR Count ← 1 TO 7
MiddayTotal ← MiddayTotal + MiddayTemperature[Count]
MidnightTotal ← MidnightTotal + MidnightTemperature[Count]
NEXT Count
MiddayAverage ← MiddayTotal/7
MidnightAverage ← MiddayTotal/7
PRINT 'The average midday temperature is ', MiddayAverage
PRINT 'The average midnight temperature is ', MidnightAverage
```

If loop not used
– total of 7 midday temperatures
– calculation of midday average (Note could be combined as one calculation,see example below)
– total of 7 midnight temperatures
– calculation of midnight average (Note could be combined as one calculation,see example below)
– output of both averages with suitable messages [5]

sample algorithm:

```
MiddayAverage ← (MiddayTemperature[1]+
MiddayTemperature[2]+MiddayTemperature[3]+
MiddayTemperature[4]+MiddayTemperature[5]+
MiddayTemperature[6]+MiddayTemperature[7])/7
MidnightAverage ← (MidnightTemperature[1]+MidnightTemperature[2]+
MidnightTemperature[3]+ Midnight[4]+Midnight[5]+ Midnight[6]+
MidnightTemperature[7])/7
PRINT 'The average midday temperature is ', MiddayAverage
PRINT 'The average midnight temperature is ', MidnightAverage
```

(c) 1 mark for the data set and 1 mark for the matching reason.
There are many possible correct answers, these are examples only.
Data set – 30, 29, 28, 31.5, 32.3, 33, 29.7
Reason – normal data that should be accepted
Data set – twenty, 23.99, seventeen, 501, –273, @#@, seventy seven
Reason – abnormal data that should be rejected [2]

(d) Maximum 6 marks in total for question part
Explanation (max 6)
– set variable called HighestMidday to a large minus number
– loop (30 or 7) times to check each midday temperature in turn
– check midday temperature against HighestMidday / midday temperature >

HighestMidday
– …replace value in HighestMidday by midday temperature
– …store array index in MiddayMonthDay/MiddayWeekday
– output HighestMidday outside the loop
– output MiddayMonthDay/MiddayWeekday outside the loop
Sample algorithm (max 4):
HighestMidday ← -999
FOR Count ← 1 TO 7
IF MiddayTemperature [Count] > HighestMidday
THEN HighestMidday ← MiddayTemperature[Count]
MiddayMonthDay/MiddayWeekday ← Count
ENDIF
NEXT Count
PRINT 'The highest midday temperature was ',HighestMidday, ' on day ', Count
If pseudocode or programming only and no explanation, then maximum 4 marks [6]

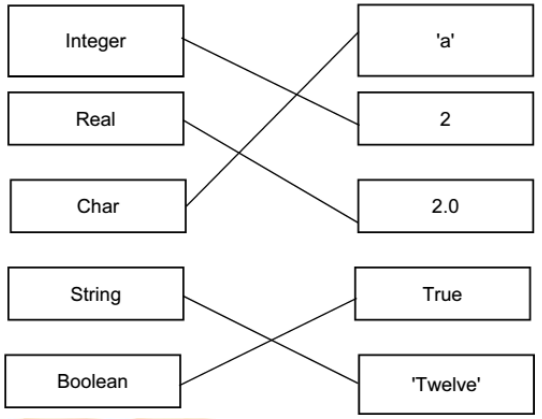
Section B

2 1 mark for each error identified + suggested correction
Line 1 or Small = 0: this should read Small = 999
line 5 or IF…: this should read IF Num < Small THEN Small = Num
line 8 or UNTIL: this should read UNTIL Counter = 10 or
UNTIL Counter > = 10 or
UNTIL Counter > 9
line 7 or PRINT…: PRINT Small should come after the end of the repeat loop
or
line 8 or UNTIL: this should come before line 7 [4]

3

Total	Reject	Weight	Output
0	0		
1.8		1.8	
	1	26.0	
8.8		7.0	
20.1		11.3	
30.1		10.0	
32.6		2.5	
	2	25.2	
37.6		5.0	
57.4		19.8	
	3	29.3	
		–1	57.4, 3

4 1 mark for each correct link, up to maximum of 4 marks



5 Any two points from
– a variable is used to store data that can change during the running of a program
– a constant is used to store data that will not be changed during the running of a program [2]

6 – FOR (… TO … NEXT)
– REPEAT (… UNTIL)
– WHILE (… DO … ENDWHILE) [3]

7 (a) – 7 [1]

(b) – Brochure No
– Uniquely identifies each property [2]

(c) Garage – Boolean
Number of Bedrooms – Number/Integer/Single
Price in $ – Number/Single/Real/Currency [3]

(d) 399000 H13
450000 H10 [2]

(e)

Field:	Property Type	Garage	Price in $	Brochure No
Table:	PROPERTY	PROPERTY	PROPERTY	PROPERTY
Sort:				
Show:	☑	☐	☑	☑
Criteria:		True	< 200000	
or:				

Summer 2015 P22

Section A

Here is a copy of the pre-release material.
DO NOT attempt Tasks 1, 2 and 3 now.
Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

Write and test a program to complete the **three** tasks.

TASK 1

A school keeps records of the weights of each pupil. The weight, in kilograms, of each pupil is recorded on the first day of term. Input and store the weights and names recorded for a class of 30 pupils. You must store the weights in a one-dimensional array and the names in another one dimensional array. All the weights must be validated on entry and any invalid weights rejected. You must decide your own validation rules. You may assume that the pupils' names are unique. Output the names and weights of the pupils in the class.

TASK 2

The weight, in kilograms, of each pupil is recorded again on the last day of term. Calculate and store the difference in weight for each pupil.

TASK 3

For those pupils who have a difference in weight of more than 2.5 kilograms, output, with a suitable message, the pupil's name, the difference in weight and whether this is a rise or a fall. Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

1 (a) All variables, constants and other identifiers should have meaningful names.

(i) Declare the array to store the pupils' names.

.....[1]

(ii) Declare the array to store the pupils' weights.

.....[1]

(iii) It has been decided to record the weights for the whole school of 600 pupils rather than one class.
Write suitable new declarations for these two arrays.

.....
.....[1]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. Use weights for **the whole school**. You should assume that Task 1 has already been completed.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[5]

(c) (i) Describe suitable validation rules for **Task 1**.

.....
.....
.....
.....[2]

(ii) Give **two** pupil weights that you could use to check the validation used in **Task 1**. Explain why you chose each weight.

Weight 1 .....
Reason for choice .....
.....
Weight 2 .....
Reason for choice .....
.....[4]

(d) Explain how you select the pupils with a fall in weight of more than 2.5 kilograms (part of **Task 3**). You may include pseudocode or programming statements as part of your explanation.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[6]

Section B

2 Read this section of program code that should input 30 positive numbers and then output the largest number input.

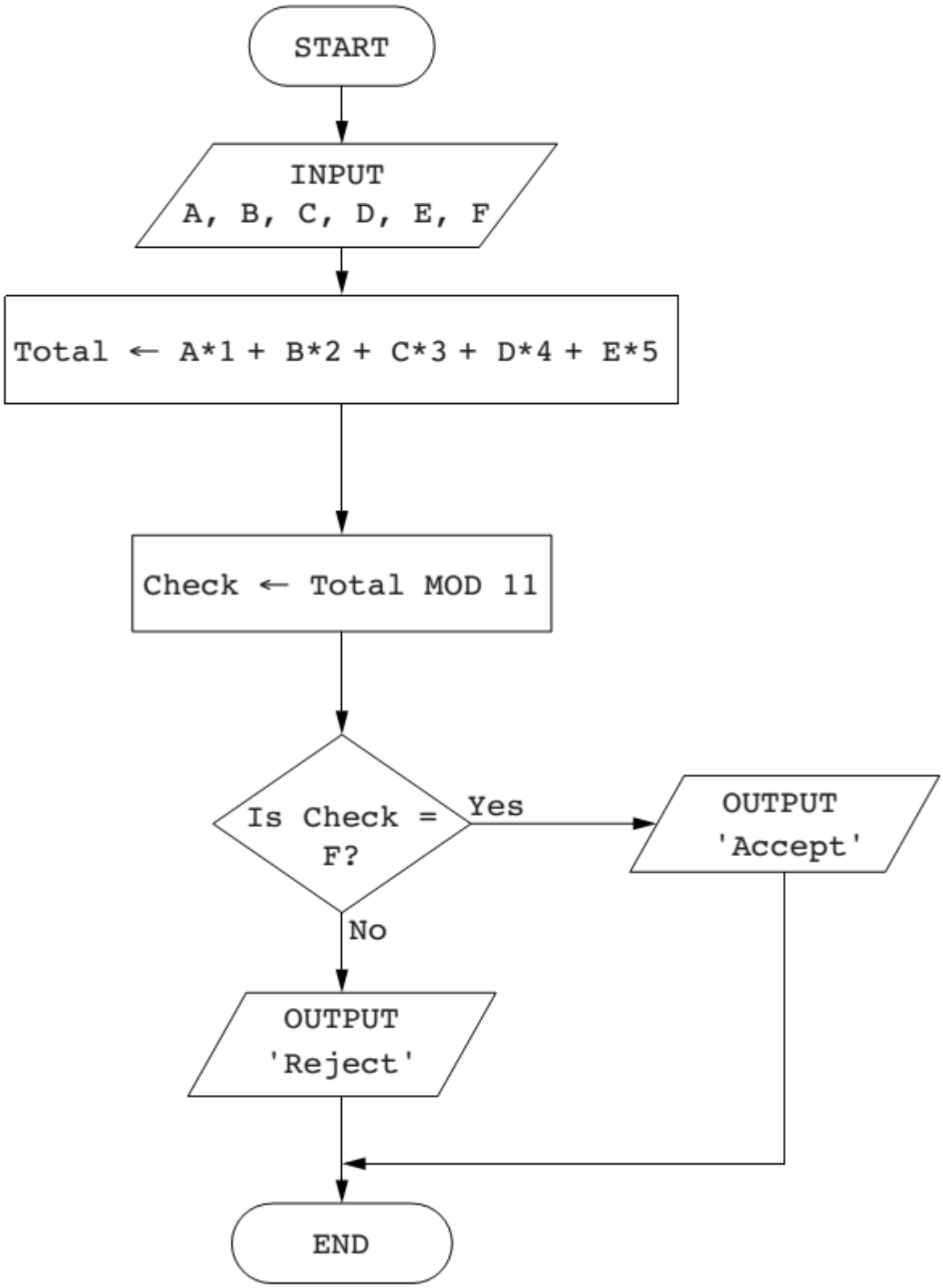
```
1 Large = 9999
2 Counter = 0
3 WHILE Counter > 30
4 DO
5 INPUT Num
6 IF Num < Large THEN Large = Num
7 Counter = Counter - 1
8 ENDWHILE
9 PRINT Large
```

There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 .....
.....
2 .....
.....
3 .....
.....
4 .....
.....[4]

3 (a) The flowchart below inputs six single digit numbers. The predefined function `MOD` gives the value of the remainder, for example, `Y ← 10 MOD 3` gives the value `Y = 1`



Complete a trace table for each of the two sets of input data.
Set 1 5, 2, 4, 3, 1, 5
Set 2 3, 2, 1, 0, 7, 3
Trace table set 1: 5, 2, 4, 3, 1, 5

A	B	C	D	E	F	Total	Check	Output

Trace table set 2: 3, 2, 1, 0, 7, 3

A	B	C	D	E	F	Total	Check	Output

(b) State the purpose of the flowchart in **part (a)**.
.....
.....[1]
(c) Identify a problem with this flowchart and explain how to correct it.
Problem .....
.....
Solution .....
.....
.....
.....[3].
4 Four programming concepts and four examples of programming code are shown below. Draw a line to link each programming concept to the correct example of programming code. [4]

Programming concept	Example of programming code
Counting	`Sum = Sum + Value[n]`
Repetition	`IF Value = 10 THEN PRINT 'X'`
Selection	`FOR Counter = 1 TO 10`
Totalling	`Amount = Amount + 1`
	`Sum = Num1 + Num2`

5 (a) Write an algorithm, using pseudocode and a `FOR … TO … NEXT` loop structure, to input 1000 numbers into an array.
.....
.....
.....
.....
.....
.....[2]

(b) Rewrite your algorithm using another loop structure.

.....
.....
.....
.....
.....
.....[4]

6 A database, MARKS, was set up to record the test results for a class of students. Part of the database is shown below.

Student Name	Class ID	Maths	English	Science	History	Geography
Paul Smith	0017	70	55	65	62	59
Ravi Gupta	0009	29	34	38	41	44
Chin Hwee	0010	43	47	50	45	52
John Jones	0013	37	67	21	28	35
Diana Abur	0001	92	88	95	89	78
Rosanna King	0016	21	13	11	27	15

(a) Give the number of fields that are in each record.

.....[1]

(b) State which fields you would choose for the primary key.

.....

Give a reason for choosing this field.

.....
.....[2]

(c) The query-by-example grid below selects all students with more than 60 marks in History or more than 60 marks in Geography.

Field:	Student Name	History	Geography
Table:	MARKS	MARKS	MARKS
Sort:	Ascending		
Show:	☑	☐	☐
Criteria:		>60	
or:			>60

Show what would be output.

.....
.....[2]

(d) Complete the query-by-example grid below to select and show the student names only of all students with less than 40 marks in both Maths and English. [3]

Field:			
Table:			
Sort:			
Show:	☐	☐	☐
Criteria:			
or:			

Marking Scheme

Section A

1 (a) (i) Many correct answers, they must be meaningful. This is an example only.
– PupilName[1:30]
or PupilName[0:29]
or PupilName[30]
or PupilName[29]
or PupilName[] [1]

(ii) Many correct answers, they must be meaningful. This is an example only.
– StartWeight[1:30]
or StartWeight[0:29]
or StartWeight[30]
or StartWeight[29]
or StartWeight[] [1]

(iii) Answers, must match (i) and (ii) above and the upper bound should have been changed
from 30 to 600 or 29 to 599 or no change if not used.
– StartWeight[1:600] or StartWeight[600]
– PupilName[1:600] or PupilName[600] [1]

(b) any four from
– prompt for entry of final weight that includes pupil's name
– input final weight
– validation check for final weight
– calculation of difference in weight
– ……using the initial weight stored in the array
– store difference in weight
(Max 4 marks)
– loop for 600 pupils
(1 mark) [5]

sample algorithm:
```
FOR Count 1 TO 600
REPEAT
PRINT 'Please enter weight for ', PupilName[Count]
INPUT FinalWeight
UNTIL FinalWeight < 120 AND FinalWeight > 20
WeightDifference[Count] FinalWeight - StartWeight[Count]
NEXT Count
```

(c) (i) any two from
– check that the weights are within a given range
– check that the weights are numeric
– check that the weights are given to one decimal point
– character/type check on name
– length check on name
[2]

(ii) 1 mark for the data and 1 mark for the matching reason.
There are many possible correct answers this is an example only.
Weight 1 – 35.2
Reason – normal data that should be accepted
Weight 2 – twenty
Reason – abnormal data that should be rejected [4]

(d) Maximum 6 marks in total for question part
Explanation (max 6)
– loop 30 or 600 times to check each difference in weight
– check for a difference in weight
– less than -2.5 (final weight – start weight) or greater than 2.5 (start weight
– final weight)
– …If so output pupil's name
– …if so output difference in weight
– …if so output message that it is a fall in weight

Sample algorithm (max 4)
```
FOR Count 1 TO 30
IF WeightDifference [Count] < -2.5
THEN PRINT PupilName[Count], 'The weight loss was ',
WeightDifference [Count]
ENDIF
NEXT Count
```
If pseudocode or programming only and no explanation, then maximum 4 marks [6]

Section B

2 1 mark for each error identified + suggested correction
Line 1 or Large =9999: this should read Large = 0
Line 3 or WHILE: this should read WHILE Counter < 30

line 6 or IF: this should read IF Num > Large THEN Large = Num
line 7 or Counter =…: this should read Counter = Counter + 1 [4]

3 (a)

Trace table set 1

A	B	C	D	E	F	Total	Check	Output
5	2	4	3	1	5	38	5	Accept

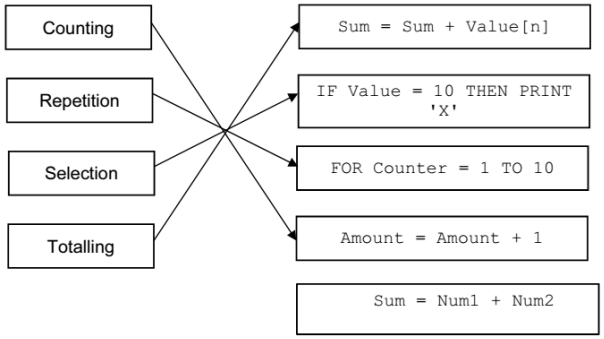
←----------(1 mark)----------→←---------(1 mark)------→

Trace table set 2

A	B	C	D	E	F	Total	Check	Output
3	2	1	0	7	3	45	1	Reject

(b) – (modulo 11) check digit calculation [1]
(c) 1 mark for identifying the problem, 2 marks for the solution
Problem – doesn't deal correctly with remainder 10/a check digit of X
Solution – check Z for X as a final digit
– have a special case where check = 10
– accept where Check = 10 and F = X [3]

4 1 mark for each correct line, two lines from one box not allowed



5 (a) 1 mark for FOR … TO … NEXT 1 mark for INPUT
FOR Count 1 TO 1000
INPUT A[Count]
NEXT (Count) [2]

(b) 4 marks
– initialisation
– start of loop
– update loop counter
– end of loop
Example1
Count 1 (1 mark)
REPEAT (1 mark)
INPUT A[Count]
Count Count + 1 (1 mark)
UNTIL Count > 1000 (1 mark)
Example2
Count 0 (1 mark)
WHILE Count < 1000 (1 mark)
DO
Count Count + 1 (1 mark)
INPUT A[Count]
ENDWHILE (1 mark)

6 (a) – 7 [1]
(b) – Class ID
– Uniquely identifies each student [2]
(c) Diana Abur, Paul Smith
– both names
– ……. correct order [2]

(d)

Field:	Student Name	Maths	English
Table:	MARKS	MARKS	MARKS
Sort:			
Show:	☑	☐	☐
Criteria:		<40	<40
or:			

(1 mark) (1 mark) (1 mark) [3]

Winter 2015 P21 & 22

Section A

Pre-release Material

Write and test a program to complete the **three** tasks.

A new born baby is kept in a cot in a hospital; the temperature of the baby is monitored every10 minutes. The temperature of the baby is recorded in degrees Celsius to one decimal place and must be within the range 36.0°C to 37.5°C.

TASK 1

To simulate the monitoring required, write a routine that allows entry of the baby's temperature in degrees Celsius. The routine should check whether the temperature is within the acceptable range, too high or too low and output a suitable message in each case.

TASK 2

Write another routine that stores the temperatures taken over a three hour period in an array. This routine should output the highest and lowest temperatures and calculate the difference between these temperatures.

TASK 3

For a baby who has a temperature difference of more than one degree Celsius, and/or has been outside the acceptable range more than twice in the three hour period, output a suitable message giving a summary of the problem.

1 (a) All variables, constants and other identifiers should have meaningful names.

(i) When you performed the tasks, you used variables.

Write suitable declarations for **two** of these.

State what you used each one for.

Variable 1 .....

Use .....

Variable 2 .....

Use .....[4]

(ii) When you performed the tasks, you may have used constants.

Write suitable declarations for **two** of these.

State what you used each one for.

Constant 1 .....

Use .....

.....

Constant 2 .....

Use .....

.....[4]

(b) Write an algorithm to complete **Task 1**, using **either** pseudocode, programming statements**or** a flowchart. [5]

(c) (i) Explain how you completed **Task 3**. You can include pseudocode or programmingstatements as part of your explanation. You should assume that Task 2 has beencompleted. [5]

(ii) Comment on the efficiency of your design for **Task 3**.

.....

.....

.....

.....[2]

Section B

2 Read this section of program code that should input 50 numbers and then output the average.

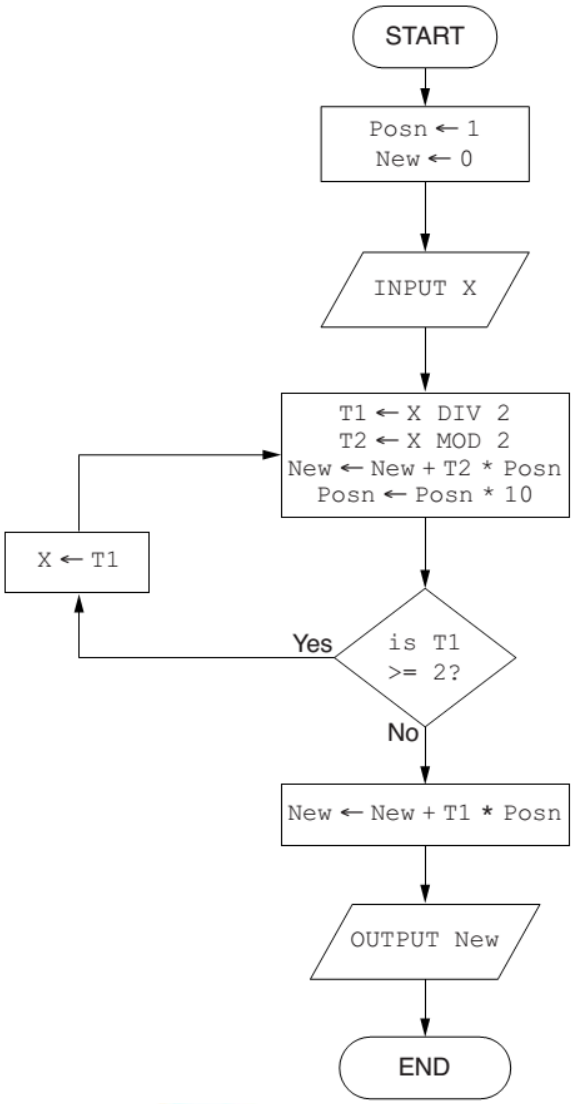
```
1 Total = 0
2 For Counter = 1 TO 50
3 INPUT Num
4 Total = Total + 1
5 Counter = Counter + 1
6 Average = Total/Counter
7 NEXT Counter
8 PRINT Average
```

There are **four** errors in this code.Locate these errors and suggest code corrections to remove each error.

1. .....
.....
2 .....
.....
3 .....
.....
4 .....
.....[4]

3 (a) The flowchart inputs an integer. The predefined function DIV gives the integer result of thedivision, e.g. Y 10 DIV 3 gives the value Y = 3. The predefined function MOD gives thevalue of the remainder, e.g. Y 10 MOD 3 gives the value Y = 1.

Complete a trace table for each of the **two** input values 5 and 12.



Trace table for input value 5

X	Posn	New	T1	T2	OUTPUT

Trace table for input value 12

X	Posn	New	T1	T2	OUTPUT

(b) State the purpose of the flowchart in **part (a)**.

.....[1]

4 A routine checks the weight of melons to be sold in a supermarket. Melons weighing under0.5 kilograms are rejected and melons weighing over 2 kilograms are also rejected. Give an example of each type of test data for this routine

Normal .....

Extreme .....

Abnormal .....[3]

5 Identify **two** different conditional statements that you can use when writing pseudocode.

1 .....

2 .....[2]

6 A picture gallery owner has decided to set up a database to keep information about the pictures he has for sale. The database table, PICTURE, will contain the following fields: Title; Artist; Description; Catalogue Number; Size (area in square centimetres); Price; Arrived (date picture arrived at gallery); Sold (whether picture is already sold)

(a) (i) State what data type you would choose for each field.

Title .....

Artist .....

Description .....

Catalogue Number .....

Size .....

Price .....

Arrived .....

Sold .....[4]

(ii) State which field you would choose for the primary key.

.....[1]

(b) Give a validation check that you can perform on each of these fields. Each validation check must be different.

Catalogue Number .....

Size .....

Price .....

Arrived .....[4]

(c) Complete the query-by-example grid below to select and show the Catalogue Number, Title and Price of all unsold pictures by the artist 'Twister'.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

Marking Scheme

Section A

1 (a) (i) Any two variables with matching uses, one mark for the variable name and one mark for the matching use. The variables and the matching uses must relate to the tasks on the exam paper. There are many possible correct answers these are examples only.

Variable 1 – Counter(: INTEGER)

Use – to use as a loop counter when entering the temperatures

Variable 2 – BabyTemperature(: REAL)

Use – to store the baby's temperature [4]

(ii) Any two constants with matching uses, one mark for the constant (name and value) and one mark for the matching use. The constants and the matching uses must relate to the tasks on the exam paper. There are several possible correct answers these are examples only.

Constant 1 – MinBabyTemperature = 36.0

Use – to keep the lowest acceptable baby temperature

Constant 2 – MaxBabyTemperature = 37.5

Use – to keep the highest acceptable baby temperature [4]

(b) Any five from

– prompt for baby's temperature

– input baby's temperature

– test for > 37.5

– … then output suitable message if this is the case

– test for < 36.0

– … then output suitable message if this is the case

– output suitable message if temperature between those values [5]

Sample algorithm:

```
PRINT 'Please enter temperature of baby '
INPUT BabyTemperature
IF BabyTemperature > MaxBabyTemperature or 37.5
THEN Print 'Temperature too high'
ELSE
IF BabyTemperature < MinBabyTemperature or 36.0
THEN Print 'Temperature too low'
ELSE Print 'Temperature OK'
ENDIF
ENDIF
```

(c) (i) Explanation

General marks award as seen

Give one mark for a mention of any one of the 4 checks below

If a mark is given for a check then mark the corresponding action taken

Maximum of five marks overall

General

– check all recorded temperatures (loop 18 times)

– update counter for those out of range

– output suitable message if counter >= 2

1 check if temperature range <= 1 and highest recorded not out of range and lowest recorded not out of range

– ... exit

2 check if temperature range > 1...

– … output suitable message e.g. "Temperature range greater than one degree"

3 check if highest recorded temperature out of range…

– … output a suitable message if at least two recorded temperatures out of range

e.g. "Temperature too high on more than one occasion"

4 check if lowest recorded temperature out of range…

– … output a suitable message if at least two recorded temperatures out of range

e.g. "Temperature too low on more than one occasion" [5]

(ii) Any two from

– only checks necessary conditions

– uses results from task 2

– checks for normal values first [2]

Section B

2 One mark for each error identified + suggested correction

line 4 or (Total =) Total + 1: this should read (Total =) Total + Num

line 5 or Counter = Counter + 1: delete this line

line 6 or (Average =)Total / Counter: swap lines 6 and 7

line 6 or (Average =)Total / Counter : this should read (Average =) Total / 50 [4]

3 (a)

Number 1 Trace table

X	Posn	New	T1	T2	Output
5	1	0			
	10	1	2	1	
2	100	1	1	0	
		101			
					101

Number 2 Trace table

X	Posn	New	T1	T2	Output
12	1	0			
	10	0	6	0	
6	100	0	3	0	
3	1000	100	1	1	
		1100			
					1100

(b) Converts a (denary) number to binary [1]

4 There are many possible correct answers this is an example only.

Normal e.g. 1.7

Extreme 0.5 or 2.0 only

Abnormal e.g. one [3]

5 – IF (… THEN … ELSE … ENDIF)

– CASE (… OF … OTHERWISE … ENDCASE) [2]

6 (a) (i) One mark for every two correct types

Title – text

Artist – text

Description – text/memo

Catalogue Number – text/(auto)number

Size – number

Price – currency/number

Arrived – date

Sold – "yes/no"/text/Boolean

0, 1 no marks

2, 3 one mark

4, 5 two marks

6, 7 three marks

8 four marks [4]

(ii) Catalogue Number [1]

(b) One mark for each correct different check

Catalogue Number Format check/Presence Check/Check Digit/Length check/uniqueness check

Size Type check/Presence Check/Range Check

Price Type check/Presence Check/Range Check

Arrived Type check/Presence Check/Range Check/Format check/Select from calendar length check [4]

(c)

Field:	Catalogue Number	Title	Price	Artist	Sold
Table:	PICTURE	PICTURE	PICTURE	PICTURE	PICTURE
Sort:					
Show:	☑	☑	☑	☐	☐
Criteria:				='Twister'	False
or:					
	(1 mark)	(1 mark)	(1 mark)	(1 mark)	(1 mark)

[5]

Winter 2015 P23

Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

DO NOT attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

Pre-release Material

Write and test a program to complete the **three** tasks.

The temperature in an apartment needs to be kept between 22°C and 24°C. This is done by the use of an automatically controlled air-conditioning system, which monitors the temperature every five minutes. The temperature of the apartment is recorded, to one decimal place, in degrees Celsius.

The cooling is activated when the temperature reaches 24.5°C and the heating is activated when the temperature reaches 21.5°C.

TASK 1

To simulate the monitoring required, write a routine that allows entry of the apartment's temperature in degrees Celsius. The routine checks whether the temperature is within the acceptable range, too high or too low and outputs a suitable message in each case.

TASK 2

Write another routine that stores, in an array, the temperatures taken over a period of five hours. This routine calculates the difference between the highest temperature and the lowest temperature. Then it outputs the highest temperature, the lowest temperature, and the difference between these temperatures.

TASK 3

Write a routine to find out how often the temperature was out of the acceptable range during the five hours and whether the temperature was too high or too low; output a suitable message showing a summary of the problem.

Your program must include appropriate prompts for the entry of data. Error messages and other outputs need to be set out clearly and understandably. All variables, constants and other identifiers must have meaningful names. Each task must be fully tested.

1 (a) All variables, constants and other identifiers should have meaningful names.

(i) When you performed the tasks, you used variables.

Write suitable declarations for **two** of these.

State what you used each one for.

Variable 1 .....

Use .....

.....

Variable 2 .....

Use .....

.....[4]

(ii) When you performed the tasks, you may have used constants.

Write suitable declarations for **two** of these.

State what you used each one for.

Constant 1 .....

Use .....

.....

Constant 2 .....

Use .....

.....[4]

(b) Write an algorithm to complete **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. You should assume that the temperatures taken over the five hours are already stored in an array.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[5]

(c) (i) Explain how you completed **Task 3**. You should assume that Task 2 has been completed. You can include pseudocode or programming statements as part of your explanation.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....[6]

(ii) Comment on the efficiency of your design for **Task 3**.

.....
.....
..... [1]

Section B

2 Read this section of program code that should input 50 numbers and then output the average of the positive numbers only.

```
1 Total = 0
2 PosCount = 0
3 FOR Counter = 1 TO 50
4 INPUT Num
5 IF Num < 0 THEN Total = Total + Num
6 IF Num > 0 THEN Counter = Counter + 1
7 Average = Total/PosCount
8 NEXT Counter
9 PRINT Num
```

There are **four** errors in this code.

Locate these errors and suggest code corrections to remove each error.

1 .....
.....

2 .....
.....
3 .....
.....
4 .....
.....[4]

3 (a) This pseudocode inputs an integer. The predefined function DIV gives the value of the division, e.g. Y 10 DIV 3 gives the value Y = 3. The predefined function MOD gives the value of the remainder, e.g. Y 10 MOD 3 gives the value Y = 1.gives the value Y = 1.

```
INPUT X
WHILE X > 15
  DO
  T1 ← X DIV 16
  T2 ← X MOD 16
  CASE T2 OF
    10:OUTPUT A
    11:OUTPUT B
    12:OUTPUT C
    13:OUTPUT D
    14:OUTPUT E
    15:OUTPUT F
    OTHERWISE OUTPUT T2
  ENDCASE
  X ← T1
ENDWHILE
CASE X OF
  10:OUTPUT A
  11:OUTPUT B
  12:OUTPUT C
  13:OUTPUT D
  14:OUTPUT E
  15:OUTPUT F
  OTHERWISE OUTPUT X
ENDCASE
```

Complete a trace table for each of the **two** input values 37 and 191.

Trace table for input value 37

X	T1	T2	OUTPUT

Trace table for input value 191

X	T1	T2	OUTPUT

(b) State the purpose of the pseudocode in **part (a)**.

.....
.....[2]

4 A routine checks the age and height of children who are allowed to enter a play area. The children must be less than 5 years of age and under 1 metre in height.

(a) The first set of test data used is age 3 and height 0.82 metres.

State what type of test data this is.

.....

Give a reason for using this test data.

.....
.....[2]

(b) Provide **two** additional sets of test data. For each, give

• the type of each set of test data
• the reason why it is used

Each type of test data and reason for use must be different.

Set 1 .....
Type .....
Reason .....
.....
.....
Set 2 .....
Type .....
Reason .....
.....
.....[6]

5 A motor boat hire company decides to set up a database to keep information about boats that are available for hire. The database table, BOAT, will contain the following fields:
Boat Name; Model; Engine Power (in hp); Number of Seats; Life Raft (whether there is a life raft kept on the boat); Day Price (price for a day's hire).

(a) Give the data type you would choose for each field.

Boat Name .....
Model .....
Engine Power .....
Number of Seats .....
Life Raft .....
Day Price .....[3]

(b) State a validation check that you can perform on each of these fields. Each validation check must be different.

Boat Name .....
Model .....
Number of Seats .....
Day Price .....[4]

(c) Complete the query-by-example grid below to select and show the Boat Name, Model and Day Price of a day's hire for all boats with 4 seats and an Engine Power of more than 100 hp.

Field:					
Table:					
Sort:					
Show:	☐	☐	☐	☐	☐
Criteria:					
or:					

[5]

Marking Scheme

Section A

1 (a) (i) Any two variables with matching uses, one mark for the variable and one mark for the matching use. The variables and the matching uses must relate to the tasks on the exam paper. There are many possible correct answers these are examples only.

Variable 1 – Counter(: INTEGER)

Use – to use as a loop counter when entering the temperatures

Variable 2 – HighestTemperature(: REAL)

Use – to store the highest temperature recorded [4]

(ii) Any two constant with matching uses, one mark for the constant and one mark for the matching use. The constants and the matching uses must relate to the tasks on the exam paper. There are several possible correct answers these are examples only.

Constant 1 – MinAppartmentTemperature = 21.5/22

Use – to keep the temperature when the air-conditioning should be switched off

Constant 2 – MaxAppartmentTemperature = 24.5/24

Use – to keep the temperature when the air-conditioning should be switched on [4]

(b) Any four from:

– initialisation, set highest apartment temperature to a low value, set lowest apartment temperature to a high value outside loop

– input temperature

– store in array

– test for temperature > highest apartment temperature reset highest apartment

temperature if this is the case

– test for temperature < lowest apartment temperature reset lowest apartment temperature

if this is the case

– calculate range

– output highest temperature, lowest temperature and the range outside loop (Max four marks)

– loop 60 times must have both tests within the loop, initialisation before the loop and

output after the loop (One mark) [5]

sample algorithm:

```
HighestTemp 0; LowestTemp 100
FOR Count 1 to 60
INPUT Temperature
ApartmentTemp[Count] Temperature
IF ApartmentTemp[Count] > HighestTemp
THEN HighestTemp ApartmentTemp[Count]
ENDIF
IF ApartmentTemp[Count] < LowestTemp
THEN LowestTemp ApartmentTemp[Count]
ENDIF
NEXT Count
Range HighestTemp – LowestTemp
PRINT 'Highest Temperature recorded ', HighestTemp
PRINT 'Lowest Temperature recorded ', LowestTemp
PRINT 'Range ', Range
```

(c) (i) Explanation six marks from:

1 – check if highest temperature <= 24 and lowest temperature >= 22…

– … message temperature always within acceptable range then exit

2 – check if highest out of range

– so count number of times temperature goes above range

– message recorded temperature too high on counted number of occasions

3 – check if lowest out of range

– so count number of times temperature goes below range

– message recorded temperature too low on counted number of occasions

General

– check all recorded temperatures (loop)

[6]

(ii) Any one from:

– only checks necessary conditions

– uses results from task 2 [1]

Section B

2 One mark for each error identified + suggested correction

line 5 or IF Num < 0: this should read IF Num > 0 (THEN Total = Total +

Num)
line 6 or (IF Num > 0) THEN Counter = Counter + 1:
this should read (IF Num > 0 THEN)Poscount = Poscount + 1
line 7 Average = Total/Poscount: this should come after the end of the repeat loop
line 9 or PRINT Num: this should read PRINT Average [4]
3 (a) Number 1 Trace Table

X	T1	T2	Output
37	2	5	5
2			2

← (1 mark) →← (1 mark) →

Number 2 Trace Table

X	T1	T2	Output
191	11	15	F
11			B

← (1 mark) →← (1 mark) →

(b) – convert a denary number to hexadecimal
– and output it in reverse order [2]
4 (a) (i) Normal
(ii) Acceptable data to test that the results are as expected. [2]
(b) One mark for the data set, one mark for the type and one mark for the matching reason
There are many possible correct answers this is an example only.
Set 1 – Age 4, height 0.9
Type – Boundary/Extreme
Reason – Data to test the validation that is just within the limits of acceptability
Set 2 – Age 10, height 1.4
Type – Abnormal
Reason – Data that should be rejected and produce an error message [6]
5 One mark for every two correct types
Boat Name – text
Model – text
Engine Power – number
Number of Seats – number
Life Raft – "yes/no"/text/Boolean
Day Price – currency/number
0, 1 no marks
2, 3 one mark
4, 5 two marks
6 three marks [3]
(b) One mark for each correct different check
Boat Name Presence Check/Type Check/Character Check
Model Format check/Type check/Presence Check/Length check/
Use of Drop-down box to select
Number of Seats Type check/Presence Check/Range Check/
Use of Drop-down box to select
Day Price Type check/Presence Check/Range Check [4]
(c)

Field:	Boat Name	Model	Day Price	Number of Seats	Engine Power
Table:	BOAT	BOAT	BOAT	BOAT	BOAT
Sort:					
Show:	☑	☑	☑	☐	☐
Criteria:				= 4	> 100
or:					
	(1 mark)	(1mark)	(1 mark)	(1 mark)	(1 mark)

[5]

PATEL